# Perceiving People over Long Periods: Algorithms, Architectures & Datasets

*Karttikeya Mangalam*

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2023-282
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2023/EECS-2023-282.html

December 15, 2023

Copyright © 2023, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Acknowledgement

My research journey has been made possible by the professors that nurtured my mind, my family who have nurtured my spirit, and the friends who have walked alongside me. Their collective wisdom and support have been the bedrock upon which my achievements rests and I am forever indebted to all of them.

Perceiving People over Long Periods: Algorithms, Architectures & Datasets

By

Karttikeya Mangalam

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Jitendra Malik, Chair
Professor Trevor Darrell
Professor Pieter Abbeel
Professor Juan Carlos Niebles

Fall 2024

Perceiving People over Long Periods: Algorithms, Architectures & Datasets

Copyright 2024
by
Karttikeya Mangalam

Abstract

Perceiving People over Long Periods: Algorithms, Architectures & Datasets

by

Karttikeya Mangalam

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Jitendra Malik, Chair

Long-form video understanding remains as one of the last enduring open problems in computer vision. While the natural world offers long periods of visual stimuli, most computer vision systems still operate within a limited temporal scope, typically just a few seconds in both input and output. This thesis presents my work developing the neural machinery, i.e., the algorithms, architectures and datasets, that extend the temporal capacity of video understanding systems to minutes and beyond.

I start by presenting my work on algorithms for long-term multimodal human motion forecasting, termed PECNet and Y-net. Next, I introduce my contributions on neural architectures for hierarchical, temporally scalable and memory-efficient neural architectures for understanding long-form videos in form of MViT and Rev-ViT. Finally, I close by presenting my work on EgoSchema, the first certifiably long-form video-language dataset, which serves as a benchmark for evaluating the long-form understanding capabilities of multimodal models. The presented benchmark results on EgoSchema highlight the existing performance gap between current state-of-the-art models and human-level long-form video understanding. I believe that my presented advancements in algorithms, architectures, and datasets not only address several existing limitations but also open new avenues for future research and application.

*To ma, papa and grandma,*
*who provided me with both my nature and nurture,*
*and unconditionally endured the results.*

# Contents

## Acknowledgments

*"If I have seen further, it is by standing on the shoulders of giants."*

— Sir Isaac Newton

My journey has been uplifted by the giants in my life: the professors that nurtured my mind, my family who have nurtured my spirit, and the friends who have walked alongside me. Their collective wisdom and support have been the bedrock of my achievements. But before I ode to the giants, I must first acknowledge the single biggest contributor to my success in the journey so far.

While I pride myself on being a diligent worker, having dedicating many Saturdays and late evenings in the lab, I am not one to be deceived by the veil of my own hard work and be fooled by randomness. My commitment to rigorous work schedules, twelve-hour days devoid of off-days when deadlines loomed, has certainly been a cornerstone of my progress. Yet, I remain cognizant that hard work alone does not paint the full picture. Indeed, an element of good fortune, even a series of fortunate events, has been a silent yet pivotal ally in my research career. In this acknowledgment, I will recognize and acknowledge everyone who contributed to my research journey but first I pay homage to the reviewing odds that have been in my favor, the serendipitous email introductions and replies that have advanced my research and the in general, the unseen hands of luck that have steered my path towards a successful completion of this thesis.

Next, I extend my heartfelt gratitude to Jitendra, my advisor, who has been an exemplary guide throughout my PhD journey. His remarkable work as a senior researcher in computer vision was already well-known to me, but working alongside him has been an experience of immeasurable value. Jitendra's infectious passion and enthusiasm for research have deeply influenced me. His ability to view my updates not just as isolated reports of progress but as steps in the long arc of scientific progress in computer vision has been enlightening. Under his mentorship, I have learned the delicate art of balancing immediate execution with long-term strategic planning towards progress on grand scientific goals. Jitendra is a veritable treasure trove of scientific trivia, quick-witted comebacks, and memorable quotes. These qualities, which I admired and enjoyed even before meeting him, have only enriched my own collection of knowledge and perspectives under his guidance. Interestingly, even before I commenced my Master's studies, Prof. Yoichi Sato, who was one of my early mentors, had cited Jitendra as a paragon of what a good PhD advisor should embody. It's been a privilege to have experienced this firsthand, and for that, I am profoundly grateful.

The Berkeley Vision Lab stands as a beacon of enchantment in the academic vision world, deriving its magic from the brilliant minds that inhabit its space. The lab's culture of collaboration is not just open but remarkably so, creating an atmosphere where smart and motivated individuals thrive. These students are always eager to engage in discussions, brainstorm new ideas, appreciate the nuances of our field, and even constructively critique each other's work. For a young, curious mind, this place is nothing short of paradise. My time here has been immensely enriching, broadening my understanding across various sub-fields of computer vision and offering the freedom to collaborate on projects that piqued my interest. The diverse and stimulating environment has been instrumental in my growth. I owe a debt of gratitude to all the computer vision professors at Berkeley — Angjoo,

Alyosha, Jitendra, and Trevor. Their dedication to nurturing such a uniquely open and invigorating research atmosphere has been pivotal in fostering not only my academic progress but also that of many others.

The unique quadfecta of Angjoo, Alyosha, Jitendra, and Trevor at Berkeley deserves further acknowledgment for their profound impact on my PhD journey. Angjoo's infectious positivity and enthusiasm have greatly uplifted the lab's atmosphere, boosting morale and fostering a supportive environment. Her leadership in organizing memorable CVPR lab dinners, especially in the later years of my study, remains a highlight. I am deeply thankful to Angjoo for her positive influence and for demonstrating the importance of work-life balance. .

Alyosha! Your enthusiasm for nature and hiking has been infectious, igniting my own passion for Berkeley's trails and the splendor of California's natural landscape. Your introduction to the hiking paths around Berkeley and the organization of lab events, including the unforgettable 'superbloom' camping trips, have created some of my most cherished social memories at the lab. Beyond these experiences, I've gained invaluable insights into the academic history of computer vision and learned to appreciate research as a journey across a long arc of time. Your influence extends further, with memorable moments that have left an indelible mark on my life. From sparking my love for Impressionist art during your CS180 course to witnessing your skillful defense of challenging positions and experiencing your profound commitment to research and scientific integrity – these instances have profoundly shaped my perspective. For all of these, and more, I am eternally grateful.

Throughout my four and a half years at the lab, Trevor's perspectives have been invaluable, often offering a well-reasoned counterbalance to the prevailing philosophical debates. His viewpoints, particularly on topics like the role of language in vision or contrasting management styles with Jitendra, have been a rich source of 'unsupervised learning' for me, from which I have greatly benefited. In the latter half of my PhD, my collaboration with Trevor's group deepened, thanks to his wholehearted support and encouragement. My PhD journey would have been significantly less rich and more monotonous without Trevor's unique contributions. I am immensely thankful for his guidance and insights.

I would also like to deeply express my gratitude towards all my amazing collaborators from across the world – Adrien, Amir, Chen Wei, Chen Zhao, Christoph, Kuan, Haoqi, Sehoon, Tushar and Yanghao for being so supporting my research through amazing ideas and discussions. I would specially like to thank Christoph, who mentored me patiently on a lot of low-level neural architecture details and has been an amazing mentor ever since.

And of course, the vision lab is nothing without its people. My journey was enriched by evolving friendships and camaraderie with my labmates and other PhD colleagues, who were a pillar of support in both my research and personal life. Special thanks to Alok and Hang for the many evenings we spent together on walks and dinners. Your company brightened countless evenings, especially during the challenging COVID times, making my experience much more enjoyable. I am also grateful to Allan, Amir, Anastasios, Ashish, Bill, Ilija, Ruilong, and Sasha for engaging me in thought-provoking and insightful discussions over the years. These interactions were invaluable in shaping my research perspective. The lab has always been a hub for top-tier postdocs, many of whom I was fortunate to have as both mentors and friends. My deepest gratitude goes to Amir Zamir, Andrea, Antonio, Assaf, Boyi, and George for their friendship and for teaching me about work culture and strategic planning for a new faculty role. A huge thank you to my travel companions, both from within and outside the lab, including Alok, Matt, Medhini, Rudy, Suzie and Tanmay. Our adventures across california coasts, mountains and forests have been unforgettable. Lastly, I extend my heartfelt thanks to Ashish, Dave, Dhruv, Ethan, Evonne, Haozhi, Jasmine, Jathushan, Neerja, Roi, Shiry, Taesung, Tim, Toru, Vickie, Vongani, Yossi, Yu, Yutong, and Zhe. From sharing hikes to casual hour long conversation to tackling deadlines together, your support and companionship have been invaluable throughout these years.

I also extend my thanks to my PhD flatmates, Ilija and Tanmay, for their remarkable patience and understanding. They graciously tolerated my untidiness and the inevitable pile-up of unwashed dishes during crunch times before deadlines. Their forbearance extended to enduring my late-night discussions and meeting noises, and on numerous occasions, they kindly safeguarded my belongings and parcels. The myriad of conversations we shared, ranging from brief chats to lengthy discussions, have significantly shaped my perspectives on a wide array of topics, both professional and personal. I am deeply appreciative of the time we spent together and the indelible impact it has had on me.

While it does too formal and awkward to me to thank the upcoming group of characters, this acknowledgement would be utterly incomplete without doing so. I extend the warmest gratitude and a virtual hug to my closest undergraduate friends — Abhinav, Amur, Ankit, and Shubh. They have been my unwavering cheerleaders and a source of strength and support over the past decade. A special mention goes to Shubh, who is celebrating his engagement on the very day I submit this thesis. Congratulations, and my heartfelt thanks for everything you've done and continue to do! The challenges of the COVID times would have been significantly harder without your presence and support.

My very sincere thanks also goes to my friend and partner, Nitya for her patience and understanding. Nitya's constant support, enduring the late nights and extended hours that my research demanded, has been invaluable. Her unwavering faith in my goals and aspirations has been, and continues to be, a source of immense strength. Her presence is a natural stress-buster, effortlessly easing my anxieties while helping me to navigate the complexities and celebrate the milestones of this journey. For her enduring support and love, I am profoundly thankful.

My deepest gratitude is reserved for my family – ma, papa, chacha, chachi, shanu, bhua and my late, grandma, who rightfully are the last authors of my life journey's papers. The love and care that Ma and Papa have showered upon their only child for my first twenty years spent in India have been boundless. Their decision to let me pursue higher studies in a distant land has been a sacrifice, one that continues to weigh heavily on their hearts. This realization of the immense opportunity cost and the longing and separation it entails has been a driving force for me, constantly motivating me to work harder and strive for success, making every sacrifice worthwhile. While words fall short of capturing the full depth of my emotions and gratitude towards my family, I believe that completing my PhD stands as a solid testament to their unconditional love and support. My family is the biggest of the giants, upon which my achievements rest!

# Chapter 1

# Introduction

Intelligent agents, be they natural and active like humans, artificial and passive such as large multimodal models, or artificial and active like social robots, must crucially rely on the ability to perceive a dynamic world populated by other moving entities. For humans, understanding others' intentions, behaviors, and goals over long periods is an innate capability [7]. However, despite more than fifty years of advancements in computer vision, with remarkably fast progress in the last decade, contemporary computer vision systems still significantly lack the ability to perceive people over long periods. While image models have experienced rapid progress across all the major divisions of computer vision – recognition [152, 112, 67], reconstruction [267, 201] and generation [94, 244], modern video understanding systems primarily operate within a limited temporal scope, typically spanning just a few seconds for both input and output.

The 2018 Turing award bestowed to Bengio, Hinton and LeCun underscores that progress in deep learning is propelled by the synergy of algorithms, architectures, and datasets. It's crucial to distinguish between algorithms and architectures: while algorithms encompass domain-specific aspects like tractable formulations and training procedures, architectures refer to broader neural network models applicable across various applications. In this thesis, I present techniques that make significant progress on all of the three fronts, algorithms, architectures and datasets, in service of improving video model capabilities to better perceive people over long periods.

First, on the algorithms front, I address the challenge of long-term human motion forecasting. Here, I will present two techniques that together enable forecasting a reliable distributional estimation of agent motion, upto a minute in future which is $10$ times longer than other the capabilities of trajectory forecasting models available at the time. Next, on the architectures front, I will first present Multiscale Vision Transformers, a technique to integrate hierarchical visual priors into transformers [270, 67]. This is succeeded by the development of reversible vision transformers, an extremely memory-efficient version of vanilla transformers. This efficiency is achieved through reversible backpropagation, allowing for scalability in transformer training across various dimensions, including longer input sequence lengths. Finally, on the dataset front, I will present EgoSchema, the *first* truly long-form video benchmark for understanding the long-term video understanding of modern multimodal models.

**Algorithms.** In Chapters 2 and 3, the focus is on algorithms tailored for human motion forecasting. This task entails predicting future positions of agents based on their observed movements over the past few seconds. To effectively tackle human motion forecasting, models need to interpret both low-level motion dynamics, like velocity and direction, and high-level cues, such as goals and motion intent. Additionally, considering the multi-agent context where multiple human trajectories interact, these models must also account for social dynamics, including collision avoidance, to enhance the accuracy of long-term forecasts.

Chapter 2 introduces PECNet, an efficient model for motion forecasting that directly captures long-term agent intent by focusing on trajectory endpoints. This model utilizes these endpoints to fill in the trajectory details for intermediate time steps. Following this, Chapter 3 presents Y-net, a scene-aware motion forecasting model that builds upon PECNet's concepts. Y-net extends endpoint conditioning to long-term forecasts, reaching up to a minute into the future. It employs intermediate waypoints for step-wise conditioning, enabling the generation of detailed and long-term forecasts. This approach also facilitates the visualization of future local distribution heat maps for the predicted locations, allowing interpretable long-term motion forecasts.

**Architectures.** In Chapter 4, I introduce the Multiscale Vision Transformers (MViT), a novel transformer architecture designed for visual inputs. MViT effectively integrates the hierarchical representation of visual priors into the traditionally uniform structure of vision transformers. This integration is grounded in appropriate visual inductive biases, and leads to significant improvements in FLOP efficiency across both videos and, with a straightforward extension, to images. Following this, Chapter 5 details the development of Reversible Vision Transformers, a highly memory-efficient variant of Vision Transformers. This model uniquely enables the memory required for transformer training activations to be independent of model depth. For example, with reversible configuration, a transformer with 100 blocks can be trained using the same amount of activation memory as one with just 2 blocks. This efficiency is achieved by re-engineering the residual connections within the transformer block, allowing it to become block-wise reversible. Building further on the MViT architecture, I propose the rev-MViT architecture, which combines the advantageous visual inductive biases with the memory-efficient training benefits of reversible transformers.

**Datasets.** Finally, in Chapter 6, I begin by clarifying the concept of long-form video tasks through the introduction of 'temporal certificates.' Temporal certificates are essentially a measure of the minimum length of a video sub-clip required for a human viewer to reliably solve a given task. Leveraging this concept, I develop a question-answer generation pipeline utilizing advanced Large Language Models (LLMs) like GPT-4. This pipeline serves as the foundation for the creation of EgoSchema, the first benchmark specifically designed for understanding long-form video content. To compile EgoSchema, I employ a rigorous process of data generation and filtering, supplemented by crowdsourcing. This method ensures a high-quality, comprehensive dataset. The final section of the chapter is dedicated to benchmarking a range of both open-source and proprietary state-of-the-art video models using EgoSchema. The results reveal a significant performance gap between these models and human capabilities, underscoring the substantial opportunity for future research in the realm of long-term video understanding.

# Chapter 2

# It is not the Journey but the Destination: Endpoint Conditioned Trajectory Prediction

## 2.1 Introduction

Predicting the movement of dynamic objects is a central problem for autonomous agents, be it humans, social robots [20], or self-driving cars [259]. Anticipation by prediction is indeed required for smooth and safe path planning in a changing environment. One of the most frequently encountered dynamic objects are humans. Hence, predicting human motion is of paramount importance for navigation, planning, human-robot interaction, and other critical robotic tasks. However, predicting human motion is nuanced, because humans are not inanimate entities evolving under Newtonian laws [14]. Rather, humans have the will to exert causal forces to change their motion and constantly adjust their paths as they navigate around obstacles to achieve their goals [328]. This complicated planning process is partially internal, and thus makes predicting human trajectories from observations challenging. Hence, a multitude of aspects should be taken into account beyond just past movement history, for instance latent predetermined goals, other moving agents in the scene, and social behavioral patterns.

In this work, we propose to address **human trajectory prediction by modeling intermediate stochastic goals we call endpoints**. We hypothesize that three separate factors interact to shape the trajectory of a pedestrian. First, we posit that pedestrians have some understanding of their long-term desired destination. We extend this hypothesis to sub-trajectories, i.e. the pedestrian has one or multiple intermediate destinations, which we define as potential endpoints of the local trajectory. These sub-goals can be more easily correlated with past observations to predict likely next steps and disentangle potential future trajectories.

This chapter is based on joint work with Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik and Adrien Gaidon [196], and is presented as it appeared in the ECCV 2020 proceedings.

Second, the pedestrian plans a trajectory to reach one of these sub-goals, taking into account the present scene elements. Finally, as the agent go about executing a plan, the trajectory gets modified to account for other moving agents, respecting social norms of interaction.

Following the aforementioned intuition, we propose to decompose the trajectory prediction problem into two sub-problems that also motivate our proposed architecture (Figure **??**). First, given the previous trajectories of the humans in the scene, we propose to estimate a latent belief distribution modeling the pedestrians' possible endpoints. Using this estimated latent distribution, we sample plausible endpoints for each pedestrian based on their observed trajectory. A socially-compliant future trajectory is then predicted, conditioned not only on the pedestrian and their immediate neighbors' histories (observed trajectories) but also everybody's estimated endpoints.

In conclusion, our contribution in this work is threefold. **First**, we propose a socially compliant, endpoint conditioned variational auto-encoder that closely imitates the multi-modal human trajectory planning process. **Second**, we propose a novel self-attention based social pooling layer that generalizes previously proposed social pooling mechanisms. **Third**, we show that our model can predict stable and plausible intermediate goals that enable setting a new state-of-the-art on several trajectory prediction benchmarks, improving by **20.9%** on SDD [229] & **40.8%** on ETH [213] & UCY [161].

## 2.2 Related work

There have been many previous studies [232] on how to forecast pedestrians' trajectories and predict their behaviors. Several previous works propose to learn statistical behavioral patterns from the observed motion trajectories [153, 175, 21, 257, 141, 6, 144, 90, 291, 154] for future trajectory prediction. Since then, many studies have developed models to account for agent interactions that may affect the trajectory — specifically, through scene and/or social information. Recently, there has been a significant focus on multi-modal trajectory prediction to capture different possible future trajectories given the past. There has also been some research on goal-directed path planning, which consider pedestrians' goals while predicting a path.

### Context-Based Prediction

Many previous studies have imported environment semantics, such as crosswalks, road, or traffic lights, to their proposed trajectory prediction scheme. Kitani et al. [150] encode agent-space interactions by a Markov Decision Process (MDP) to predict potential trajectories for an agent. Ballan et al. [15] leverage a dynamic Bayesian network to construct motion dependencies and patterns from training data and transferred the trained knowledge to testing data. With the great success of the deep neural network, the Recurrent Neural Network (RNN) has become a popular modeling approach for sequence learning. Kim et al. [145] train a RNN combining multiple Long Short-term Memory (LSTM) units to predict the location of nearby cars. These approaches incorporate rich environment cues from the RGB image of the scene for pedestrians' trajectory forecasting.

Behaviour of surrounding dynamic agents is also a crucial cue for contextual trajectory prediction. Human behavior modeling studied from a crowd perspective, *i.e.*, how a pedestrian interacts with other pedestrians, has also been studied widely in human trajectory prediction literature. Traditional approaches use *social forces* [118, 199, 298, 2] to capture pedestrians' trajectories towards their goals with attractive forces, while avoiding collisions in the path with repulsive forces. These approaches require hand-crafted rules and features, which are usually complicated and insufficiently robust for complicated high-level behavior modeling. Recently, many studies applied Long Short Term Memory (LSTM [123]) networks to model trajectory prediction with the social cues. Alahi et al. [3] propose a Social LSTM which learns to predict a trajectory with joint interactions. Each pedestrian is modeled by an individual LSTM, and LSTMs are connected with their nearby individual LSTMs to share information from the hidden state.

## Multimodal Trajectory Prediction

In [158, 106], the authors raise the importance of accounting for the inherent multi-modal nature of human paths *i.e.*, given pedestrians' past history, there are many plausible future paths they can take. This shift of emphasis to plan for multiple future paths has led many recent works to incorporate multi-modality in their trajectory prediction models. Lee et al. [158] propose a conditional variational autoencoder (CVAE), named DESIRE, to generate multiple future trajectories based on agent interactions, scene semantics and expected reward function, within a sampling-based inverse optimal control (IOC) scheme. In [106], Gupta et al. propose a Generative Adversarial Network (GAN) [94] based framework with a novel social pooling mechanism to generate multiple future trajectories in accordance to social norms. In [233], Sadeghian et al. also propose a GAN based framework named SoPhie, which utilizes path history of all the agents in a scene and the scene context information. SoPhie employs a social attention mechanism with physical attention, which helps in learning social information across the agent interactions. However, these socially-aware approaches do not take into account the pedestrians' ultimate goals, which play a key role in shaping their movement in the scene. A few works also approach trajectory prediction via an inverse reinforcement learning (IRL) setup. Zou et al. [329] applies Generative Adversarial Imitation Learning (GAIL) [121] for trajectory prediction, named Social-Aware GAIL (SA-GAIL). With IRL, the authors model the human decision-making process more closely through modeling humans as agents with states (past trajectory history) and actions (future position). SA-GAIL generates socially acceptable trajectories via a learned reward function.

## Conditioned-on-Goal

Goal-conditioned approaches are regarded as inverse planning or *prediction by planning* where the approach learns the final intent or goal of the agent before predicting the full trajectory. In [224], Rehder *et al*. propose a particle filtering based approach for modeling destination conditioned trajectory prediction and use explicit Von-Mises distribution based probabilistic framework for prediction. Later in a follow-up work, [225] Rehder *et al*. further propose a deep learning based destination estimation approach to tackle intention recognition and trajectory prediction simultaneously.

The approach uses fully Convolutional Neural Networks (CNN) to construct the path planning towards some potential destinations which are provided by a recurrent Mixture Density Network (RMDN). While both the approaches make an attempt for destination conditioned prediction, a fully probabilistic approach trains poorly due to unstable training and updates. Further, they ignore the presence of other pedestrians in the scene which is key for predicting shorter term motions which are missed by just considering the environment. Rhinehart et al. [227] propose a goal-conditioned multi-agent forecasting approach named PRECOG, which learns a probabilistic forecasting model conditioned on drivers' actions intents such as ahead, stop, etc. However, their approach is designed for vehicle trajectory prediction, and thus conditions on semantic goal states. In our work, we instead propose to utilize destination position for pedestrian trajectory prediction.

In [165], Li *et al*. posit a Conditional Generative Neural System (CGNS), the previous established state-of-the-art result on the ETH/UCY dataset. They propose to use variational divergence minimization with soft-attention to predict feasible multi-modal trajectory distributions. Even more recently, Bhattacharyya *et al*. [23] propose a conditional flow VAE that proposed a general normalizing flow for structured sequence prediction and applies it to the problem of trajectory prediction. Concurrent to our work, Deo *et al*. [59] propose P2TIRL, a Maximum Entropy Reinforcement Learning based trajectory prediction module over a discrete grid. The work [23] shares state-of-the-art with [59] on the Stanford Drone Dataset (SDD) with the TrajNet [234] split. However, these works fail to consider the human aspect of the problem, such as interaction with other agents. We compare our proposed PECNet with all three of the above works on both the SDD & ETH/UCY datasets.

## 2.3 Proposed Method

In this work, we aim to tackle the task of human trajectory prediction by reasoning about all the humans in the scene jointly while also respecting social norms. Suppose a pedestrian $p^k$ enters a scene $\mathcal{I}$. Given the previous trajectory of $p$ for past $t_p$ steps, as a sequence of coordinates $\mathcal{T}_p^k := \{\mathbf{u}^k\}_{i=1}^{t_p} = \{(x^k, y^k)\}_{i=1}^{t_p}$, the problem requires predicting the future position of $p^k$ on $\mathcal{I}$ for next $t_f$ steps, $\mathcal{T}_f^k := \{\mathbf{u}^k\}_{i=t_p+1}^{t_p+t_f} = \{(x, y)\}_{i=t_p+1}^{t_p+t_f}$.

As mentioned in Section 4.1, we break the problem into two daisy chained steps. First, we model the sub-goal of $p^k$, i.e. the last observed trajectory points of $p^k$ say, $\mathcal{G}^k = \mathbf{u}^k|_{t_p+t_f}$ as a representation of the predilection of $p^k$ to go its pre-determined route. This sub-goal, also referred to as the endpoint of the trajectory, the pedestrian's desired end destination for the current sequence. Then in the second step, we jointly consider the past histories $\{\mathcal{T}_p^k\}_{k=1}^{\alpha}$ of all the pedestrians $\{p^k\}_{k=1}^{\alpha}$ present in the scene and their estimated endpoints $\{\mathcal{G}^k\}_{k=1}^{\alpha}$ for predicting socially compliant future trajectories $\mathcal{T}_f^k$. In the rest of this section we describe in detail, our approach to achieve this, using the endpoint estimation VAE for sampling the future endpoints $\mathcal{G}$ and a trajectory prediction module to use the sampled endpoints $\hat{\mathcal{G}}^k$ to predict $\mathcal{T}_f$.

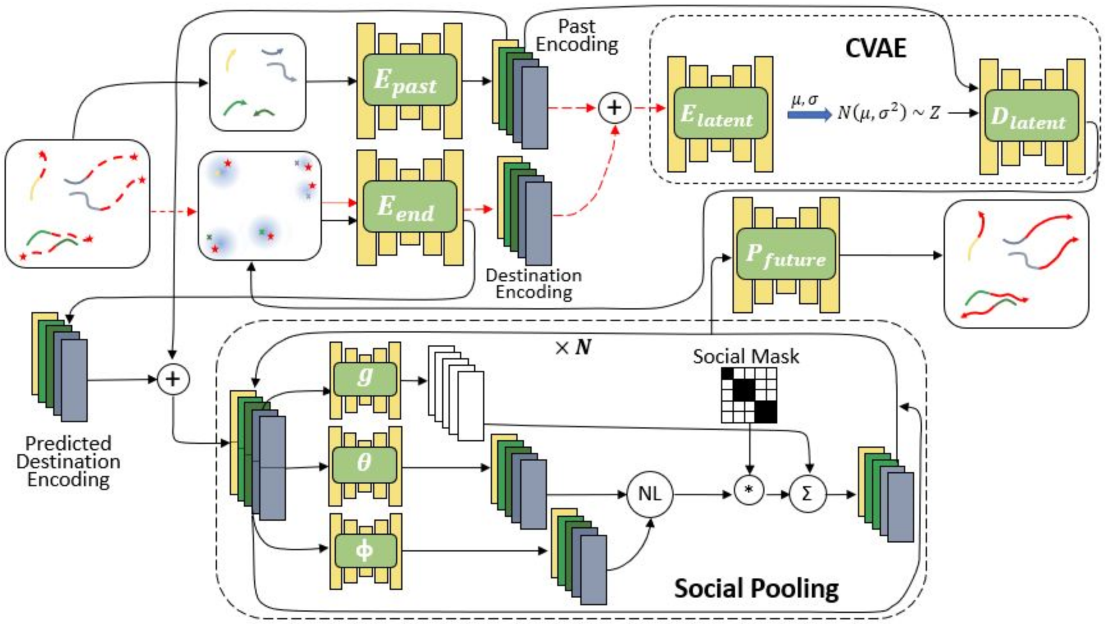



Figure 2.1: **Architecture of PECNet**: PECNet uses past history, $\mathcal{T}_i$ along with ground truth endpoint $\mathcal{G}_c$ to train a VAE for multi-modal endpoint inference. Ground-truth endpoints are denoted by $\star$ whereas **x** denote the sampled endpoints $\hat{\mathcal{G}}_c$. The sampled endpoints condition the social-pooling & predictor networks for multi-agent multi-modal trajectory forecasting. Red connections denote the parts utilized only during training. Shades of the same color denote spatio-temporal neighbours encoded with the block diagonal social mask in social pooling module. Further Details in Section 2.3.

## Endpoint VAE

We propose to model the predilection of the pedestrian as a sub-goal endpoint $\mathcal{G} := \mathbf{u}_{t_f} = (x_{t_f}, y_{t_f})$ which is the last observed trajectory point for pedestrian $p^k$. First, we infer a distribution on $\mathcal{G}$ based on the previous location history $\mathcal{T}_i$ of $p^k$ using the Endpoint VAE.

As illustrated in Figure 3.2, we extract the previous history $\mathcal{T}_i^k$ and the ground truth endpoint $\mathcal{G}^k$ for all pedestrian $p_k$ in the scene. We encode the past trajectory $\mathcal{T}_i^k$ of all $p_k$ independently using a past trajectory encoder $\mathbf{E}_{past}$. This yields us $\mathbf{E}_{past}(\mathcal{T}_i)$, a representation of the motion history. Similarly, the future endpoint $\mathcal{G}^k$ is encoded with an Endpoint encoder $\mathbf{E}_{end}$ to produce $\mathbf{E}_{end}(\mathcal{G}^k)$ independently for all $k$. These representations are concatenated together and passed into the latent encoder $\mathbf{E}_{latent}$ which produces parameter $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ for encoding the latent variable $z = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$ of the VAE. Finally, we sample possible latent future endpoints from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$, concatenate it with $\mathbf{E}_{past}(\mathcal{T}_i)$ for past context and decode using the latent decoder $\mathbf{D}_{latent}$ to yield our guesses for $\hat{\mathcal{G}}^k$. Since the ground truth $\mathcal{G}^k$ belongs to the future, and is unavailable at test time, during evaluation we sample $z$ from $\mathcal{N}(0, \sigma_T\mathbf{I})$, concatenate with $\mathbf{E}_{past}(\mathcal{T}_i)$ (as done in training) and then use the learned

$\mathbf{D}_{latent}$ to estimate the future $\hat{\mathcal{G}}^k$. This is illustrated in Figure 3.2 where the red connections are only used in the training and not in the evaluation phase.

**Truncation trick**: In [25], Brock *et al*. introduce the 'Truncation Trick' as a method of trade-off between the fidelity and variety of samples produced by the generator in BigGAN. In this work, we propose an analogous trick for evaluation phase in multi-modal trajectory forecasting where the variance of the latent endpoint sampling distribution is changed according to the number of samples ($K$) allowed for multi-modal prediction. In a situation requiring few shot multi-modal prediction, such as under computation constraints, where only a few samples ($K = 1, 2$ or $3$) are permissible, we propose to use $\sigma_T = 1$ and truncate the sampling distribution at $\pm c\sqrt{K-1}$. In contrast, in situations where a high number of predictions are to be generated (such as $K = 20$, a standard setting on benchmarks) we propose to use $\sigma_T > 1$ with no truncation. We posit that this procedure allows simple adjustment of prediction diversity in favor of overall performance for different $K$, thereby providing a simple method of achieving good performance in all settings without requiring any retraining.

## Endpoint conditioned Trajectory Prediction

Using the sampled estimate of the *endpoints* $\hat{\mathcal{G}}$ from Endpoint VAE, we employ the endpoint encoder $\mathbf{E}_{end}$ once again (within the same forward pass) to obtain encodings for the sampled endpoints $\mathbf{E}_{end}(\hat{\mathcal{G}}^k)$. This is used along with prediction network to *plan* the path $\mathcal{T}_f$ starting to $\mathcal{G}$ thereby predicting the future path.

Note that, another design choice could have been that even during training, the ground truth $\mathbf{E}_{end}(\mathcal{G}^k)$ are used to predict the future $\mathcal{T}_f$. This seems reasonable as well since it provides cleaner, less noisy signals for the downstream social pooling & prediction networks while still training the overall module end to end (because of coupling through $\mathbf{E}_{past}$). However, such a choice will decouple training of the Endpoint VAE (which would then train only with KL Divergence and AWL loss, refer Section 2.3) and social pooling network (which would then train only with ATL loss, refer 2.3) leading to inferior performance empirically.

The sampled endpoints' representations $\mathbf{E}_{end}(\hat{\mathcal{G}}^k)$ are then concatenated with corresponding $\mathbf{E}_{past}(\mathcal{T}_i)$ (as in Section 2.3) and passed through $N$ rounds of social pooling using a social pooling mask $\mathbf{M}$ for all the pedestrians in the scene jointly. The social pooling mask $\mathbf{M}$ is $\alpha \times \alpha$ block diagonal matrix denoting the social neighbours for all $\{p_i\}_{i=1}^{\alpha}$ pedestrians in the scene. Mathematically,

$$\mathbf{M}[i,j] = \begin{cases} 0 & \text{if } \min_{1 \le m,n \le t_p} \|\mathbf{u}_m^i - \mathbf{u}_n^j\|_2 > t_{dist} \\ 0 & \text{if } \min_{1 \le m \le t_p} |\mathcal{F}(\mathbf{u}_0^i) - \mathcal{F}(\mathbf{u}_m^j)| * \min_{1 \le m \le t_p} |\mathcal{F}(\mathbf{u}_m^i) - \mathcal{F}(\mathbf{u}_0^j)|) > 0 \\ 1 & \text{otherwise} \end{cases} \tag{2.1}$$

where $\mathcal{F}(.)$ denoted the actual frame number the trajectory was observed at. Intuitively, $\mathbf{M}$ defines the spatio-temporal neighbours of each pedestrian $p_i$ using proximity threshold $t_{dist}$ for distance in space and ensure temporal overlap. Thus, the matrix $\mathbf{M}$ encodes crucial information regarding social locality of different trajectories which gets utilized in attention based pooling as described below.

**Social Pooling**: Given the concatenated past history and sampled way-point representations $X_k^{(1)} = (\mathbf{E}_{past}(\mathcal{T}_p^k), \mathbf{E}_{end}(\hat{\mathcal{G}}^k))$ we do $N$ rounds of social pooling where the $(i+1)$th round of pooling recursively updates the representations $X_k^{(i)}$ from the last round according to the non-local attention mechanism [278]:

$$X_k^{(i+1)} = X_k^{(i)} + \frac{1}{\sum_{j=1}^{\alpha} \mathbf{M}_{ij} \cdot e^{\boldsymbol{\phi}(X_k^{(i)})^T \boldsymbol{\theta}(X_j^{(i)})}} \sum_{j=1}^{\alpha} \mathbf{M}_{ij} \cdot e^{\boldsymbol{\phi}(X_k^{(i)})^T \boldsymbol{\theta}(X_j^{(i)})} \mathbf{g}(X_k^{(i)}) \tag{2.2}$$

where $\{\boldsymbol{\theta}, \boldsymbol{\phi}\}$ are encoders of $X_k$ to map to a learnt latent space where the representation similarity between $p_i$ and $p_j$ trajectories is calculated using the embedded gaussian $\exp(\boldsymbol{\phi}(X_k)^T \boldsymbol{\theta}(X_j))$ for each round of pooling. The social mask, $\mathbf{M}$ is used point-wise to allow pooling only on the spatio-temporal neighbours masking away other pedestrians in the scene. Finally, $\boldsymbol{g}$ is a transformation encoder for $X_k$ used for the weighted sum with all other neighbours. The whole procedure, after being repeated $N$ times yields $X_k^{(N)}$, the pooled prediction features for each pedestrian with information about the past positions and future destinations of all other neighbours in the scene.

Our proposed social pooling is a novel method for extracting relevant information from the neighbours using non-local attention. The proposed social non local pooling (S-NL) method is permutation *invariant* to pedestrian indices as a useful inductive bias for tackling the social pooling task. Further, we argue that this method of learnt social pooling is more robust to social neighbour mis-identification such as say, mis-specified distance ($t_{dist}$) threshold compared to previously proposed method such as max-pooling [106], sorting based pooling [233] or rigid grid-based pooling [3] since a learning based method can ignore spurious signals in the social mask $\mathbf{M}$.

The pooled features $X_k^{(N)}$ are then passed through the prediction network $\mathbf{P}_{future}$ to yield our estimate of rest of trajectory $\{\mathbf{u}^k\}_{k=t_p+1}^{t_p+t_f}$ which are concatenated with sampled endpoint $\hat{\mathcal{G}}$ yields $\hat{\mathcal{T}}_f$. The complete network is trained end to end with the losses described in the next subsection.

## Loss Functions

For training the entire network end to end we use the loss function,

$$\mathcal{L}_{PECNet} = \lambda_1 \underbrace{D_{KL}(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}) \| \mathcal{N}(0, \mathbf{I}))}_{\text{KL Div in latent space}} + \lambda_2 \underbrace{\|\hat{\mathcal{G}}_c - \mathcal{G}_c\|_2^2}_{AEL} + \underbrace{\|\hat{\mathcal{T}}_f - \mathcal{T}_f\|^2}_{ATL} \tag{2.3}$$

where the KL divergence term is used for training the Variational Autoencoder, the Average endpoint Loss (AEL) trains $\mathbf{E}_{end}$, $\mathbf{E}_{past}$, $\mathbf{E}_{latent}$ and $\mathbf{D}_{latent}$ and the Average Trajectory Loss (ATL) trains the entire module together.

| | Network Architecture |
|---|---|
| $\mathbf{E}_{end}$ | $2 \to 8 \to 16 \to 16$ |
| $\mathbf{E}_{past}$ | $16 \to 512 \to 256 \to 16$ |
| $\mathbf{E}_{latent}$ | $32 \to 8 \to 50 \to 32$ |
| $\mathbf{D}_{latent}$ | $32 \to 1024 \to 512 \to 1024 \to 2$ |
| $\boldsymbol{\phi}, \boldsymbol{\theta}$ | $32 \to 512 \to 64 \to 128$ |
| $\mathbf{g}$ | $32 \to 512 \to 64 \to 32$ |
| $\mathbf{P}_{predict}$ | $32 \to 1024 \to 512 \to 256 \to 22$ |

Table 2.1: Network architecture details for all the sub-networks used in the module.

## 2.4 Experiments

### Datasets

**Stanford Drone Dataset**: Stanford Drone Dataset [229] is a well established benchmark for human trajectory prediction in bird's eye view. The dataset consists of 20 scenes captured using a drone in top down view around the university campus containing several moving agents like humans and vehicles. It consists of over $11,000$ unique pedestrians capturing over $185,000$ interactions between agents and over $40,000$ interactions between the agent and scene [229]. We use the standard test train split as used in [233, 106, 59] and other previous works.
**ETH/UCY**: Second is the ETH [213] and UCY [161] dataset group, which consists of five different scenes – ETH & HOTEL (from ETH) and UNIV, ZARA1, & ZARA2 (from UCY). All the scenes report the position of pedestrians in world-coordinates and hence the results we report are in metres. The scenes are captured in unconstrained environments with few objects blocking pedestrian paths. Hence, scene constraints from other physical non-animate entities is minimal. For bench-marking, we follow the commonly used leave one set out strategy i.e., training on four scenes and testing on the fifth scene [233, 106, 165].

### Implementation Details

All the sub-networks used in proposed module are Multi-Layer perceptrons with ReLU non-linearity. Network architecture for each of the sub-networks are mentioned in Figure 2.1. The entire network is trained end to end with the $\mathcal{L}_{E\text{-}VAE}$ loss using an ADAM optimizer with a batch size of $512$ and learning rate of $3 \times 10^{-4}$ for all experiments. For the loss coefficient weights, we set $\lambda_1 = \lambda_2 = 1$. We use $N = 3$ rounds of social pooling for Stanford Drone Dataset and $N = 1$ for ETH & UCY scenes. Using social masking, we perform the forward pass in mini-batches instead of processing all the pedestrians in the scene in a single forward pass (to aboid memory overflow) constraining all the neighbours of a pedestrian to be in the same mini-batch.
**Metrics**: For prediction evaluation, we use the Average Displacement Error (ADE) and the Final Displacement Error (FDE) metrics which are commonly used in literature [106, 3, 2, 165]. ADE

| | SoPhie | S-GAN | DESIRE | CF-VAE* | P2TIRL† | SimAug† | O-S-TT | O-TT | Ours | PECNet (Ours) |
|---|---|---|---|---|---|---|---|---|---|---|
| K | 20 | 20 | 5 | 20 | 20. | 20 | 20 | 20 | 5 | 20 |
| ADE | 16.27 | 27.23 | 19.25 | 12.60 | 12.58 | 10.27 | 10.56 | 10.23 | 12.79 | **9.96** |
| FDE | 29.38 | 41.44 | 34.05 | 22.30 | 22.07 | 19.71 | 16.72 | 16.29 | 25.98 | **15.88** |

Table 2.2: Comparison of our method against several recently published multi-modal baselines and previous state-of-the-art method (denoted by *) on the Stanford Drone Dataset [229]. '-S' & '-TT' represents ablations of our method without social pooling & truncation trick. We report results for in pixels for both $K = 5$ & 20 and for several other $K$ in Figure 2.4. † denotes concurrent work. Lower is better.

is the average $\ell_2$ distance between the predictions and the ground truth future and FDE is the $\ell_2$ distance between the predicted and ground truth at the last observed point. Mathematically,

$$ADE = \frac{\sum_{j=t_i+1}^{t_p+t_f+1} \|\hat{\mathbf{u}}_j - \mathbf{u}_j\|_2}{t_f} \qquad FDE = \|\hat{\mathbf{u}}_{t_p+t_f+1} - \mathbf{u}_{t_p+t_f+1}\|_2 \tag{2.4}$$

where $\mathbf{u}_j$, $\hat{\mathbf{u}}_j$ are the ground truth and our estimated position of the pedestrian at future time step $j$ respectively.

**Baselines**: We compare our PECNet against several published baselines including previous state-of-the-art methods briefly described below.

- Social GAN (S-GAN) [106]: Gupta *et al*. propose a multi-modal human trajectory prediction GAN trained with a variety loss to encourage diversity.
- SoPhie [233]: Sadeghian *et al*. propose a GAN employing attention on social and physical constraints from the scene to produce human-like motion.
- CGNS [165]: Li *et al*. posit a Conditional Generative Neural System (CGNS) that uses conditional latent space learning with variational divergence minimization to learn feasible regions to produce trajectories. They also established the previous state-of-the-art result on the ETH/UCY datasets.
- DESIRE [158]: Lee *et al*. propose an Inverse optimal control based trajectory planning method that uses a refinement structure for predicting trajectories.
- CF-VAE [23]: Recently, a conditional normalizing flow based VAE proposed by Bhattacharyya *et al*. pushes the state-of-the-art on SDD further. Notably, their method also does not also rely on the RGB scene image.
- P2TIRL [59]: A concurrent work by Deo *et al*. proposes a method for trajectory forecasting using a grid based policy learned with maximum entropy inverse reinforcement learning policy. They closely tie with the previous state-of-the-art [23] in ADE/FDE performance.

- SimAug [173]: More recently, a concurrent work by Liang *et al*. proposes to use additional 3D multi-view simulation data adversarially, for novel camera view adaptation. [173] improves upon the P2TIRL as well, with performance close to PECNet's base model. However our best model (with pooling and truncation) still achieves a better ADE/FDE performance.

- Ours-TT: This represents an ablation of our method without using the truncation trick. In other words, we set $\sigma_T$ to be identically 1 for all $K$ settings. Truncation trick ablations with different $K$ are shown in Fig 2.4 & Table 2.2.

- Ours-S-TT: This represents an ablation of our method without using both the social pooling module and the truncation trick *i.e*. the base PECNet. We set $\sigma_T = 1$ and $N = 0$ for the number of rounds of social pooling and directly transmit the representations to $\mathbf{P}_{future}$, the prediction sub-network.

## Quantitative Results

In this section, we compare and discuss our method's performance against above mentioned baselines on the ADE & FDE metrics.

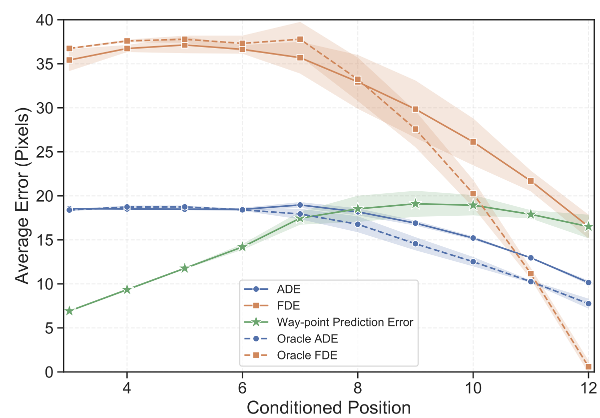



Figure 2.2: **Conditioned Way-point positions & Oracles**: We evaluate the performance of the proposed method against the choice of future conditioning position on ADE & FDE metrics. Further, we evaluate the performance of a destination oracle version of the model that receives perfect information on conditioned position for predicting rest of the trajectory.

**Stanford Drone Dataset**: Table 2.2 shows the results of our proposed method against the previous baselines & state-of-the-art methods. Our proposed method achieves a superior performance compared to the previous state-of-the-art [23, 59] on both ADE & FDE metrics by a significant margin of **20.9%**. Even without using the proposed social pooling module & truncation trick (OUR-S-TT), we achieve a very good performance (10.56 ADE), underlining the importance of future endpoint conditioning in trajectory prediction.

As observed by the difference in performance between Ours-S-TT and Our-TT, the social pooling module also plays a crucial role, boosting performance by 0.33 ADE ($\sim$ 2.1%). Note that, while both P2TIRL [59] & SimAug [173] are concurrent works, we compare with their methods' performance as well in Table 2.2 for experimental comprehensiveness. All reported results averaged for 100 trials.

| | S-GAN | | SoPhie | | CGNS* | | S-LSTM | | Ours - S - TT | | PECNet (Ours) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ADE | FDE | ADE | FDE | ADE | FDE | ADE | FDE | ADE | FDE | ADE | FDE |
| ETH | 0.81 | 1.52 | 0.70 | 1.43 | 0.62 | 1.40 | 1.09 | 2.35 | 0.58 | 0.96 | **0.54** | **0.87** |
| HOTEL | 0.72 | 1.61 | 0.76 | 1.67 | 0.70 | 0.93 | 0.79 | 1.76 | 0.19 | 0.34 | **0.18** | **0.24** |
| UNIV | 0.60 | 1.26 | 0.54 | 1.24 | 0.48 | 1.22 | 0.67 | 1.40 | 0.39 | 0.67 | **0.35** | **0.60** |
| ZARA1 | 0.34 | 0.69 | 0.30 | 0.63 | 0.32 | 0.59 | 0.47 | 1.00 | 0.23 | 0.39 | **0.22** | **0.39** |
| ZARA2 | 0.42 | 0.84 | 0.38 | 0.78 | 0.35 | 0.71 | 0.56 | 1.17 | 0.24 | 0.35 | **0.17** | **0.30** |
| AVG | 0.58 | 1.18 | 0.54 | 1.15 | 0.49 | 0.97 | 0.72 | 1.54 | 0.32 | 0.54 | **0.29** | **0.48** |

Table 2.3: Quantitative results for various previously published methods and state-of-the-art method (denoted by *) on commonly used trajectory prediction datasets. Both ADE and FDE are reported in metres in world coordinates. 'Our-S-TT' represents ablation of our method without social pooling & truncation trick.

**ETH/UCY**: Table 3.2 shows the results for evaluation of our proposed method on the ETH/UCY scenes. We follow the leave-one-out evaluation protocol with $K = 20$ as in CGNS [165]/Social-GAN [106]. All reported numbers are *without* the truncation trick. In this setting too, we observe that our method outperforms previously proposed methods, including the previous state-of-the-art [165]. We push the state-of-the-art on average by $\sim$ **40.8%** with the effect being the most on HOTEL ($74.2\%$) and least on ETH ($12.9\%$). Also, without the social pooling & truncation trick (OUR-S-TT) the performance is still superior to the state-of-the-art by $34.6\%$, underlining the usefulness of conditioning on the endpoint in PECNet.

**Conditioned Way-point positions & Oracles**: For further evaluation of our model, we condition on future trajectory points other than the last observed point which we refer to as *way-points*. Further, to decouple the errors in inferring the conditioned position from errors in predicting a path to that position, we use a destination (endpoint) oracle. The destination oracle provides ground truth information of the conditioned position to the model, which uses it to predict the rest of the trajectory. All of the models, with and without the destination oracle are trained from scratch for each of the conditioning positions. Referring to Figure 2.2, we observe several interesting and informative trends that support our earlier hypotheses. (A) As a sanity check, we observe that as we condition on positions further into the future, the FDE for both the Oracle model & the proposed model decrease with a sharp trend after the 7th future position. This is expected since points further into the future provide more information for the final observed point. (B) The ADE error curves for both the oracle and the proposed model have the same decreasing trend albeit with a gentler slope than FDE because the error in predicting the other points (particularly the noisy points in the middle of the trajectory) decreases the gradient.

(C) Interestingly, our model's ADE and FDE is not significantly different from that of the Oracle model for points close in the future and the error in the two models are approximately the same until about the 7th future position. This suggests that till around the middle of the future, the conditioned way-points do not hold significant predictive power on the endpoint and hence using our noisy guesses vs. the oracle's ground truth for their position does not make a difference.

**Way-point Prediction Error**: The way-point position error is the $\ell_2$ distance between the prediction of location of the conditioned position and its ground truth location (in the future). Referring to Figure 2.2, we observe an interesting trend in the way-point error as we condition on points further into the future. The way-point prediction error increases at the start which is expected since points further into the future have a higher variance. However, after around the middle (7th point) the error plateaus and then even slightly decreases. This lends support to our hypothesis that pedestrians, having predilection towards their destination, exert their will towards it. Hence, *predicting the last observed way-point allows for lower prediction error* than way-points in the middle! This in a nutshell, confirms the motivation of this work.

**Effect of Number of samples (K)**: All the previous works use $K = 20$ samples (except DESIRE which uses $K = 5$) to evaluate the multi-modal predictions for metrics ADE & FDE. Referring to Figure 2.4, we see the expected decreasing trend in ADE & FDE with time as $K$ increases. Further, we observe that our proposed method achieves the same error as the previous works with much smaller $K$. Previous state-of-the-art achieves $12.58$ [59] ADE using $K = 20$ samples which is matched by PECNet at half the number of samples, $K = 10$. This further lends support to our hypothesis that conditioning on the inferred way-point significantly reduces the modeling complexity for multi-modal trajectory forecasting, providing a better estimate of the ground truth. Lastly, as $K$ grows large ($K \rightarrow \infty$) we observe that the FDE slowly gets closer to $0$ with more number of samples, as the ground truth $\mathcal{G}_c$ is eventually found. However, the ADE error is still large $(6.49)$ because of the errors in the rest of the predicted trajectory. This is in accordance with the observed ADE $(8.24)$ for the oracle conditioned on the last observed point (*i.e.* $0$ FDE error) in Fig. 2.2.

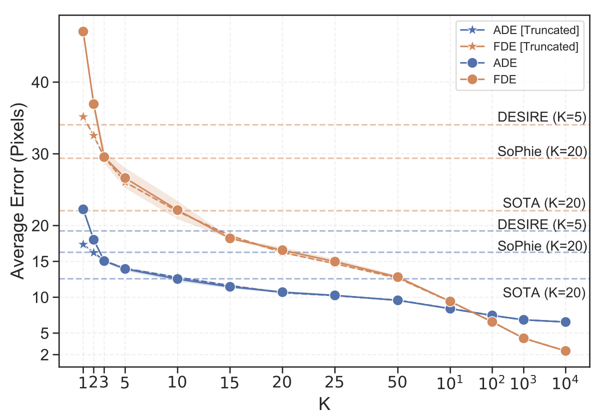



Figure 2.4: **Performance across** $K$: ADE & FDE performance of our method against number of samples used for evaluation. Several previous baselines are mentioned as well with their number of samples used. Our method significantly outperforms the state-of-the-art reaching their performance with much lesser number of samples & performing much better with same number of samples as theirs ($K = 20$).

**Design choice for VAE**: We also evaluate our design choice of using the inferred future way-points $\hat{\mathcal{G}}_c$ for training subsequent modeules (social pooling & prediction) instead of using the ground truth $\mathcal{G}_c$. As mentioned in Section 2.3, this is also a valid choice for training PECNet end to end. Empirically, we find that such a design achieves $10.87$ ADE and $17.03$ FDE. This is worse ($\sim 8.8\%$) than using $\hat{\mathcal{G}}_c$ which motivates our design choice for using $\hat{\mathcal{G}}_c$ (Section 2.3).

**Truncation Trick**: Fig. 2.4 shows the improvements from the truncation trick for an empirically chosen hyperparameter $c \approx 1.2$. As expected, small values of $K$ gain the most from truncation, with the performance boosting from 22.85 ADE ($48.8$ FDE) to $17.29$ ADE ($35.12$ FDE) for $K = 1$ ($\sim 24.7\%$).

## 2.5 Conclusion

In this work we present PECNet, a pedestrian endpoint conditioned trajectory prediction network. We show that PECNet predicts rich & diverse multi-modal socially compliant trajectories across a variety of scenes. Further, we perform extensive ablations on our design choices such as endpoint conditioning position, number of samples & choice of training signal to pinpoint achieved performance gains. We also introduce a “truncation trick" for trajectory prediction, a simple method for adjusting diversity for performance in trajectory prediction without retraining. Finally, we benchmark PECNet across multiple datasets including Stanford Drone Dataset [229], ETH [213] & UCY [161] in all of which PECNet achieved state-of-the-art performance.

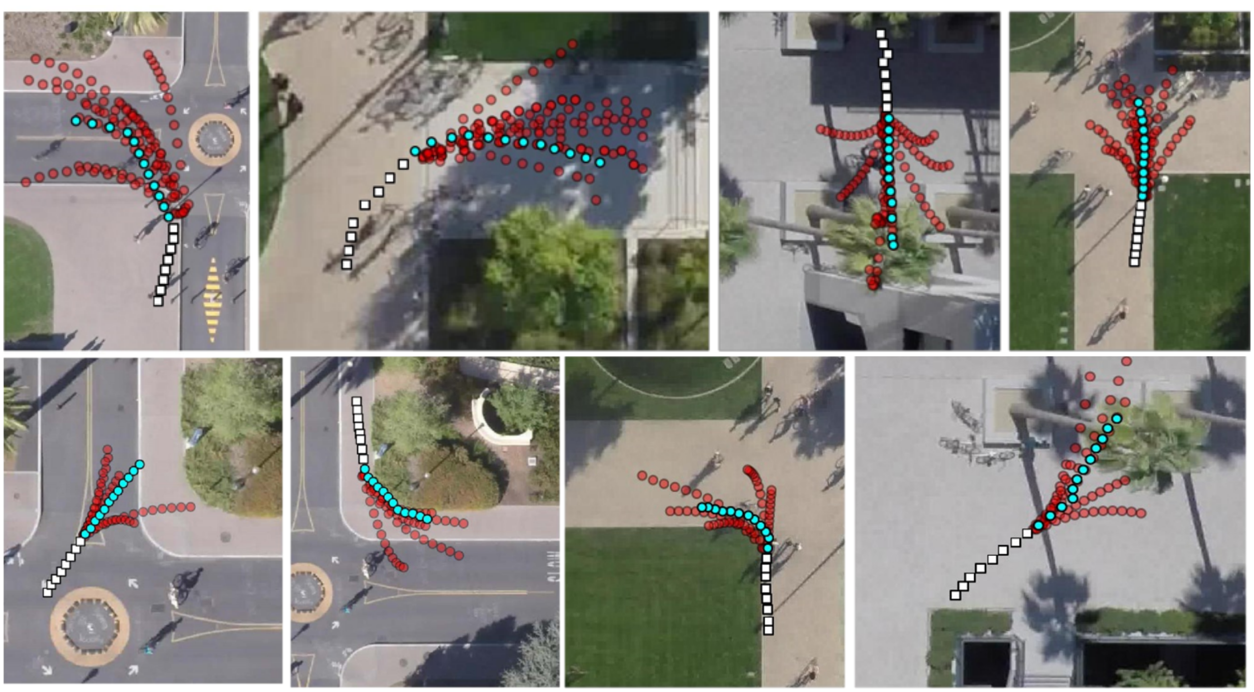


Figure 2.3: **Visualizing Multimodality**: We show visualizations for some multi-modal and diverse predictions produced by PECNet. White represents the past $3.2$ seconds while red & cyan represents predicted & ground truth future respectively over next $4.8$ seconds. Predictions capture a wide-range of plausible trajectory behaviours while discarding improbable ones like, endpoints opposite to pedestrian’s direction of motion.

# Chapter 3

# From Goals, Waypoints & Paths To Long Term Human Trajectory Forecasting

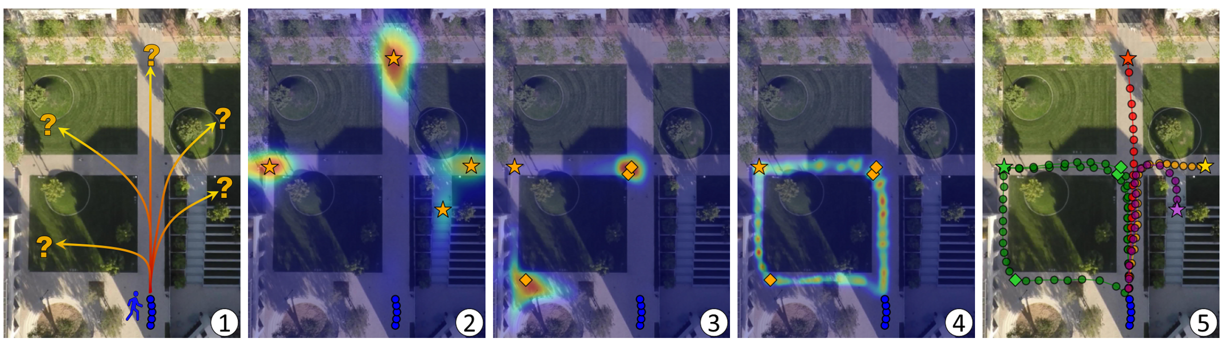


Figure 3.1: We tackle the problem of long term human trajectory forecasting. Given the past motion of an agent (blue) on a scene over the last five seconds, we aim to predict the multimodal future motion over the next minute ①. To achieve this, we propose factorizing overall multimodality into its *epistemic* and *aleatoric* factors. The *epistemic* factor is modeled with an estimated distribution over the long term goals ② while the *aleatoric* factor is modeled as a distribution over the intermediate waypoints ③ and trajectory ④ for each goal separately. This is repeated for multiple goals and waypoints for scene-compliant multimodal human trajectory forecasting ⑤. Each color indicates predicted trajectories for a different sampled goal.

## 3.1 Introduction

Sequence prediction is a fundamental problem in several engineering disciplines such as signal processing, pattern recognition, control engineering, and in virtually any domain concerned with

This chapter is based on joint work with Yang An, Harshayu Girase and Jitendra Malik [195], and is presented as it appeared in the ICCV 2021 proceedings.

temporal measurements. From the seminal work of A. A. Markov [198] on predicting the next syllable in the poem *Eugene Onegin* with Markov chains, to modern day autoregressive descendants like GPT-3 [27], next element prediction in a sequence has a long standing history. Time series forecasting is a key instantiation of the sequence prediction problem in the setting where the sequence is formed by elements sampled in time. Several classic techniques such as Autoregressive Moving Average Models (ARMA) [281] have been incorporated in deep learning architectures [269, 123] in modern day state-of-the-art time series forecasting methods [238].

However, humans are not inanimate Newtonian entities, slaves to predetermined physical laws and forces. Predicting the future motion of a billiard ball smoothly rolling on a pool table under friction and physical constraints is a problem of different nature from forecasting human motion and positions. Humans are goal conditioned agents that, unlike the ball, exert their will through actions to achieve a desired outcome [261]. Anticipating human motion is of fundamental importance to dynamic agents such as other humans, autonomous robots [20] and self-driving vehicles [260]. Human motion is inherently goal directed and is put in place by the agent to bring about a desired effect.

Nevertheless, even conditioned on the agent's past motion and overarching long term goals, is the future trajectory deterministic? Consider yourself standing at a crossing on a busy street, waiting for the pedestrian light to turn green. While you have every intention of crossing the street, the exact future trajectory remains stochastic as you might swerve to avoid other pedestrians, speed up your pace if the light is about to turn red, or pause abruptly if an unruly cyclist dashes by. Hence, even conditioned on the past observed motion and scene semantics, future human motion is inherently stochastic [117] owing to both *epistemic* uncertainty caused by latent decision variables like long term goals and *aleatoric* variability [60] stemming from random decision variables such as environmental factors. This dichotomy is even sharper in long term forecasting since due to the increased uncertainty in the future, the aleatoric randomness influences the trajectory much more strongly in long rather than short temporal horizons.

This motivates a factorized multimodal approach for human dynamics modeling where both factors of stochasticity are modeled hierarchically rather than lumped jointly. We hypothesize that the long term latent goals of the agent represent the *epistemic* uncertainty within motion prediction. While the agent has a goal in mind while planning and executing their trajectory, this is unknown to the prediction system. In physical terms, this is akin to the question of *where* the agent wants to go. Similarly, the *aleatoric* uncertainty is expressed in the stochasticity of the path leading to the goal, which encompasses factors like environment variables such as other agents, partial scene information available to the agent and most importantly, the unconscious randomness in human decisions [142]. In physical terms, this is akin to the question of *how* the agent reaches the goal.

Hence, we propose to model the *epistemic* uncertainty first and then model the *aleatoric* stochasticity conditioned on the obtained estimate. Concretely, with the RGB scene and the past motion history, we first estimate an explicit probability distribution over the agent's long term goals. This represents the *epistemic* uncertainty in the prediction system. We also estimate distributions over a few chosen future waypoint positions which along with the sampled goal points are used to obtain explicit probability maps over all the remaining intermediate trajectory positions. This represents the aleatoric uncertainty in the prediction system. Together the samples from the *epistemic*

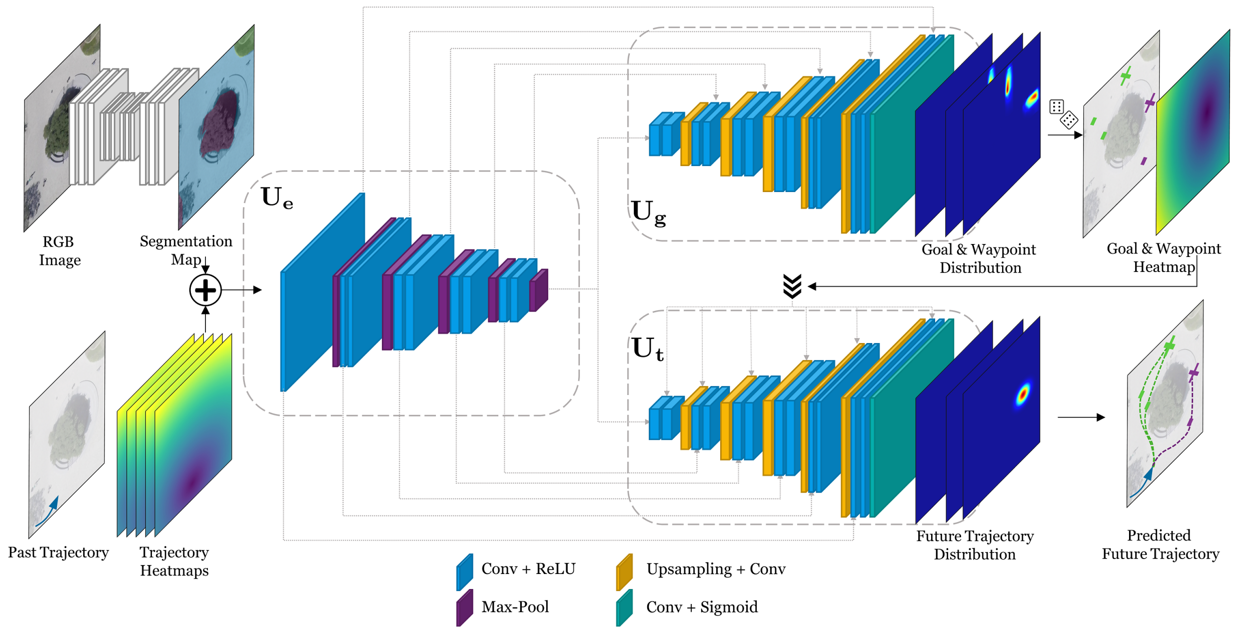



Figure 3.2: **Model Architecture**: Y-net comprises of three sub-networks $\mathbf{U}_e$, $\mathbf{U}_g$ and $\mathbf{U}_t$ modeled after the U-net architecture [230] (Section 3.3). Y-net adopts a factorized approach to multimodality, expressing the stochasticity in goals and waypoints through estimated distributions furnished by $\mathbf{U}_g$. And multimodality in paths is achieved through estimated probability distributions obtained by $\mathbf{U}_t$ conditioned on samples from $\mathbf{U}_g$ for predicting diverse multimodal scene-compliant futures.

goal distribution and the *aleatoric* waypoint and trajectory distribution form the predicted future trajectory.

In summary, our contribution is threefold. **First**, we propose a novel long term prediction setting that extends up to a minute in the future which is about an order of magnitude longer than previous literature. **Second**, we propose Y-net, a scene-compliant long term trajectory prediction network that explicitly models both the *goal* and *path* multi-modalities while making effective use of the scene semantics. **Third**, we show that the factorized multimodality modeling enables Y-net to improve the state-of-the-art both on the proposed long term settings and the well-studied short term prediction settings. We benchmark Y-net's performance on the Stanford Drone [228] and the ETH [214]/UCY[161] benchmark in the short term setting. It outperforms previous approaches by significant margins of 13.0% in ADE and 31.7% in FDE metric on SDD, and on-par in ADE and by 7.4% in FDE on ETH/UCY. Further, we also study Y-net's performance in the proposed long term prediction setting on the Stanford Drone and the Intersection Drone Dataset [24] where it substantially improves the performance of state-of-the-art short term methods by over 50.7% and 39.7% respectively, on ADE and 77.1% and 56.0% respectively, on FDE metric. The preprocessed data, model, and code can be found here for future work: `https://karttikeya.github.io/publication/ynet/`

## 3.2 Related Works

Several recent studies have investigated human trajectory prediction in different settings. Broadly, these approaches can be grouped based on the proposed formulation for multimodality in forecasting, inputs signals available to the prediction model and the nature and form of prediction results furnished by the model. Several diverse input signals such as agent's past motion history [118], human pose [194], RGB scene image [106, 233, 32, 158, 174], scene semantic cues [32], location [236, 165, 23] and gaze of other pedestrian [194, 297] in the scene, moving vehicles such as cars [236] and also latent inferred signals such as agent's goals [196] have been used. The form of prediction results produced are also diverse with multimodality [174] and scene-compliant forecasting being central to the prior works.

**Unimodal Forecasting**: Early trajectory forecasting work focused on unimodal predictions of the future. Social Forces [118] proposes modeling interactions as attractive and repulsive forces and future trajectory as a deterministic path evolving under these forces. Social LSTM [3] focuses on other agents in the scene and models their effects through a novel pooling module. [297] forecasts motion in ego-centric views and exploits body pose and gaze along with camera wearer's ego-motion for other agent's future location prediction. [271] proposes to use attention to model target agent's interaction with other agent's. [194] predicts trajectory as the 'global' branch for pose prediction and proposes to condition downstream tasks such as pose prediction on predicted unimodal trajectories.

**Multimodality through Generative Modeling**: Another line of work aims to model the stochasticity inherent in future prediction through a latent variable with a defined prior distribution through approaches such as conditional variational auto-encoders [148]. DESIRE [158] is an inverse reinforcement learning based approach that uses multimodality in sampling of a latent variable that is ranked and optimized with a refinement module. [194] introduces the use of a CVAE for capturing multimodality in the final position of the pedestrians conditioned on the past motion history. Trajectron++ [236] represents agent's trajectories in a graph structured recurrent network for scene compliant trajectory forecasting, taking into account the interaction with a diverse set of agents. LB-EBM [209] learns an energy-based model in the latent space and a policy generator to map the latent vector into a trajectory. The attention based method AgentFormer [311] jointly models the time dimension and social interactions using a sequence representation while preserving each agent's identity. Introvert [239] uses a 3D visual attention mechanism conditioned on the observed trajectory to extract scene and social information from videos. A future displacement distribution is predicted and multiple sequences can be sampled.

A different line of work includes Social GAN [106] which uses adversarial losses [94] for incorporating multimodality in predictions. While such generative approaches do produce diverse trajectories, overall coverage of critical modes cannot be guaranteed and little control is afforded over the properties of predicted trajectories such as direction, number of samples, *etc*. In contrast, our method, Y-net, estimates explicit probability maps which allow easily incorporating spatial constraints for a downstream task.

**Multimodality through spatial probability estimates**: Another line of work obtains multimodality via estimated probability maps. Activity Forecasting from Kitani *et al*. [150] proposes to use a hidden Markov Decision process for modeling the future paths. However, in contrast to our work, the future predictions in [150] are conditioned on activity labels such as 'approach car', 'depart car', *etc*. More recently, some works have used a grid based scene representation to estimate probabilities for future time steps [173, 174, 59]. Relatedly, some prior works such as [194, 319, 32] propose a goal-conditioned trajectory forecasting method. However, no prior works have proposed factorized modeling of *epistemic* uncertainty or goals and *aleatoric* uncertainty or paths as Y-net uses.

## 3.3 Proposed Method

The problem of multimodal trajectory prediction can be formulated formally as follows. Given a RGB scene image $\mathcal{I}$ and past positions of a pedestrian in the scene $\mathcal{I}$ denoted by $\{\mathbf{u}_n\}_{n=1}^{n_p}$ for the past $t_p = n_p/\texttt{FPS}$ seconds sampled at the frame rate $\texttt{FPS}$, the model aims to predict the position of the pedestrian for the next $t_f$ seconds in the future, denoted by $\{\mathbf{u}_n^i\}_{n=n_p+1}^{n_p+n_f}$ where $t_f = n_f/\texttt{FPS}$. Since the future is stochastic, multiple predictions for the future trajectories are produced. In this work, we factorize the overall stochasticity into two modes. First are the modes relating to *epistemic* uncertainty *i.e*. multimodality in the final destination for which the module produces $K_e$ goals. Second are the modes relating to the *aleatoric* uncertainty *i.e*. multimodality in the path taken to the destination stemming from uncontrolled randomness given the goal, for which the module produces $K_a$ predictions for each estimated goal. In the short temporal horizon limit, since the overall path length is small, the options for paths to a given goal are limited and similar to each other. This is naturally modeled by constraining $K_a = 1$ and so the total number of paths predicted ($K$ in prior works) is the same as $K_e$ in the short horizon setting. However, for longer temporal horizons, there are several paths to the same goal and hence $K_a > 1$. Next, we describe in detail the working of our model, Y-net and its three sub-networks $\mathbf{U}_e$, $\mathbf{U}_g$ and $\mathbf{U}_t$ followed by details of the non-parametric sampling process (Section 3.3) and loss functions used.

### Y-net Sub-Networks

To effectively use scene information in semantic space (image-like) with trajectory information (coordinates), pixel-wise alignment needs to be created between the different modalities. Some prior works [233] achieve this by encoding the RGB image $\mathcal{I}$ as a hidden state vector extracted from a pretrained CNN network. While this provides the network with scene information, any meaningful spatial signal gets highly conflated when flattened into a vector and pixel alignment is destroyed. This is highlighted in [196] establishing previous state-of-the-art without any scene information, underlining the misuse of image information in prior works. In this work, we adopt a trajectory-on-scene heatmap representation that solves the alignment issue by representing the trajectory in the same space as image $\mathcal{I}$.

| | S-GAN | CF-VAE | P2TIRL | SimAug | PECNet | LB-EBM | Y-net (Ours) | DESIRE | TNT | PECNet | Y-net (Ours) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $K = 20$ | | | | | | | $K = 5$ | | | |
| ADE | 27.23 | 12.60 | 12.58 | 10.27 | 9.96 | 8.87 | **7.85** | 19.25 | 12.23 | 12.79 | **11.49** |
| FDE | 41.44 | 22.30 | 22.07 | 19.71 | 15.88 | 15.61 | **11.85** | 34.05 | 21.16 | 29.58 | **20.23** |

Table 3.1: **Short temporal horizon forecasting results on SDD**: Our method significantly outperforms previous state-of-the-art methods on the Stanford Drone Dataset [229] on both the ADE and FDE metrics for both settings of $K$, where $K$ represents the number of multimodal samples . Reported errors are in pixels with $t_p = 3.2$ sec, $t_f = 4.8$ sec, $n_p = 8, n_f = 12$ and lower is better.

### Trajectory-on-Scene Heatmap Representation

The RGB image $\mathcal{I}$ is first processed with a semantic segmentation network such as a U-net [230] that produces segmentation map $\mathbf{S}$ of $\mathcal{I}$ comprising of $C$ classes determined according to the affordance provided by the surface to an agent for actions such as walking, standing, running etc. In a parallel branch, the past motion history $\{\mathbf{u}_n\}_{n=1}^{n_p}$ is converted to a trajectory heatmap $\mathbf{H}$ of spatial sizes of $\mathcal{I}$ and $n_p$ channels with one channel for each timestep. Mathematically,

$$\mathbf{H}(n, i, j) = 2\frac{\|(i, j) - \mathbf{u}_n\|}{\max\limits_{(x,y)\in\mathcal{I}} \|(x, y) - \mathbf{u}_n\|}$$

The heatmap trajectory representation is then concatenated with the semantic map $\mathcal{S}$ along the channel dimension producing the trajectory-on-scene heatmap tensor $\mathbf{H}_\mathcal{S}$ a $H \times W \times (C + n_p)$ dimensional input tensor which is passed to the encoder network $\mathbf{U}_e$.

### Trajectory-on-Scene Heatmap Encoder $\mathbf{U}_e$

The tensor $\mathbf{H}_\mathcal{S}$ is processed with the encoder $\mathbf{U}_e$ designed as a U-net encoder [230] (Fig. 3.2). The encoder $\mathbf{U}_e$ consists of $M$ blocks, where the spatial dimensions are reduced from $H \times W$ to $H_M \times W_M$ halving after every block using max pooling (stride 2) and the channel depth is increased sequentially from $C + n_p$ to $C_M$ doubling after a certain number of blocks using convolutional layers with ReLU. The final spatially compact and deep representation after block $M$ along with the $M - 1$ intermediate tensors $\mathbf{H}_m$ with $1 \leq m \leq M$ are passed onto the goal decoder $\mathbf{U}_g$ and the trajectory decoder $\mathbf{U}_t$ as discussed below.

### Goal and Waypoint Heatmap Decoder $\mathbf{U}_g$

The processed trajectory-on-scene tensors $\mathbf{H}_m$ at various spatial resolutions are passed onto the goal and waypoint heatmap decoder $\mathbf{U}_g$ which is modeled after the expansion arm in the U-net architecture [230]. A center block consisting of two convolutional layers with ReLU first takes in the final and spatially compact feature tensor $\mathbf{H}_M$. Then the expansion arm spatially doubles the resolution at the beginning of every block using bilinear up-sampling and convolution

| | ETH | HOTEL | UNIV | ZARA1 | ZARA2 | AVG |
|---|---|---|---|---|---|---|
| S-GAN | 0.81/1.52 | 0.72/1.61 | 0.60/1.26 | 0.34/0.69 | 0.42/0.84 | 0.58/1.18 |
| PECNet | 0.54/0.87 | 0.18/0.24 | 0.35/0.60 | 0.22/0.39 | 0.17/0.30 | 0.29/0.48 |
| LB-EBM | 0.30/0.52 | 0.13/0.20 | 0.27/0.52 | 0.20/0.37 | 0.15/0.29 | 0.21/0.38 |
| Introvert | 0.42/0.70 | 0.11/0.17 | 0.20/0.32 | 0.16/0.27 | 0.16/0.25 | 0.21/0.34 |
| Trajectron++ | 0.39/0.83 | 0.12/0.21 | **0.20**/0.44 | **0.15**/0.33 | **0.11**/0.25 | 0.19/0.41 |
| AgentFormer | **0.26**/0.39 | 0.11/0.14 | 0.26/0.46 | **0.15/0.23** | 0.14/0.24 | **0.18**/0.29 |
| Y-net (Ours) | 0.28/**0.33** | **0.10/0.14** | 0.24/**0.41** | 0.17/0.27 | 0.13/**0.22** | **0.18/0.27** |

Table 3.2: **Short term forecasting results on ETH/UCY benchmark**: Our proposed method establishes new state-of-the-art results on both the ADE/FDE metrics on the popular ETH-UCY benchmark with standard short-horizon settings (same as SDD). Reported errors are in meters and lower is better.

(together forming Deconvolution [230]). After every Deconvolution, the corresponding intermediate representation $\mathbf{H}_m$ from $\mathbf{U}_e$ is fused using skip connections and the features are processed with two convolutional layers with ReLU non-linearity. Merging intermediate high-resolution feature maps from $\mathbf{U}_e$ is necessary since just using the final feature $\mathbf{H}_M$ would severely limit the final resolution of the goal heatmap, thus missing fine spatial details that are preserved in the intermediate feature maps. The U-net block starts with a deconvolution operation followed by feature merging and two convolutional layers, all of which is repeated sequentially $M$ times to form $\mathbf{U}_g$. The output layer consists of a convolutional layer followed by a pixelwise sigmoid that for each $N^w$ chosen waypoint $\mathbf{u}_{w_i}$ and the goal $\mathbf{u}_{n_p+n_f}$ produces an explicit, non-parametric probability distribution, $P(\mathbf{u}_{w_i})$ and $P(\mathbf{u}_{n_p+n_f})$ after normalization. The overall output shape of $\mathbf{U}_g$ is $H \times W \times (N^w + 1)$. Thus, for each $N^w$ waypoint plus the goal, this submodule predicts a $H \times W$ matrix, where the $(i, j)$th element of the matrix represents the estimate probability value of the agent being at location $(i, j)$ at the selected timestep.

### Trajectory Heatmap Decoder $\mathbf{U}_t$

$\mathbf{U}_t$ comprises of $M$ decoder blocks which proceed in a similar fashion as $\mathbf{U}_g$ (Section 3.3). However, in contrast to $\mathbf{U}_g$, $\mathbf{U}_t$ is conditioned on the sampled goal and waypoints in addition to the scene $\mathbf{S}$ and past trajectory $\{\mathbf{u}_n\}_{n=1}^{n_p}$. The probability distributions estimated by $\mathbf{U}_g$ are used to sample potential goal and waypoints sets. The sampling process is described in Section 3.3 and further details are in Supplementary Section 1. In total, $K_e$ goals are sampled and for each goal $K_a$ waypoint sets are sampled representing $K_a$ paths to the same goal. The obtained coordinate sample sets $\hat{\mathbf{u}}_{n_p+n_f}$ for the goal and $\{\hat{\mathbf{u}}_{w_i}\}_{i=1}^{N^w}$ for the intermediate waypoints are converted to a heatmap representation $\mathbf{H}_g$ similar to the past trajectory as described in Section 3.3. Finally, the obtained goal and waypoint conditioning tensor $\mathbf{H}_g$ is downsampled to fit the spatial size of each corresponding block and along with the corresponding $\mathbf{H}_m$ is concatenated to the output of the previous $\mathbf{U}_t$ block and passed into the next block. For each future timestep, it predicts a separate probability distribution, resulting in an output of shape $H \times W \times n_f$, with each channel corresponding to the location distribution in

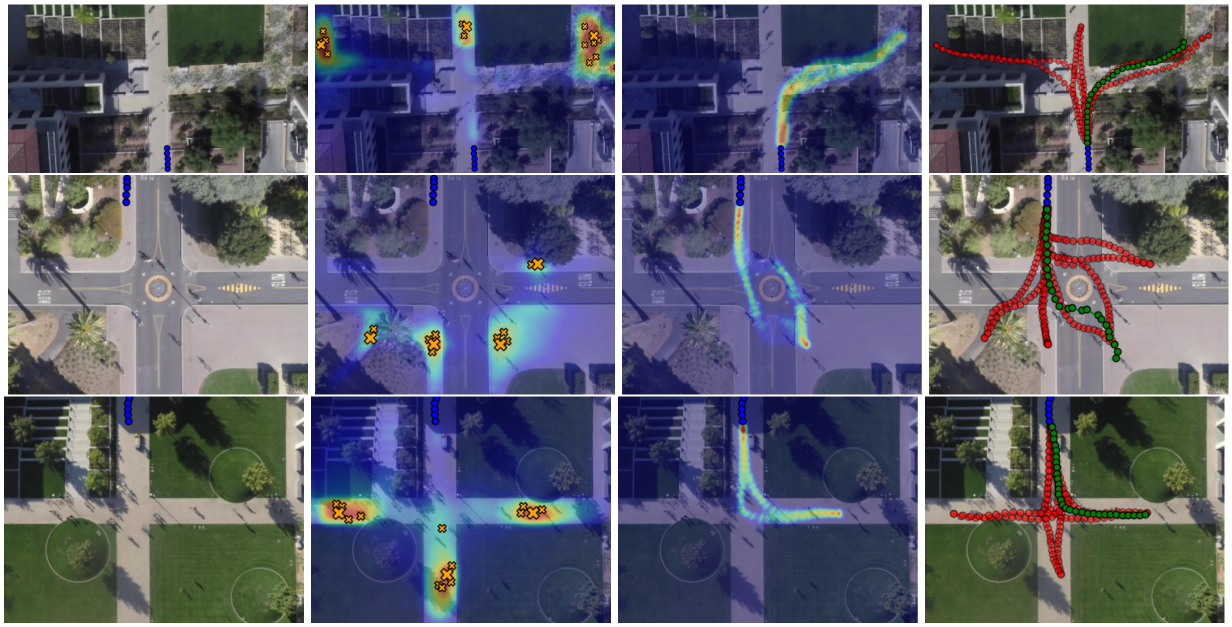


Figure 3.3: **Qualitative Long Term Trajectory Forecasting Results**: We show various heatmaps and visualizations for three different scenes (rows) in SDD testset. The first column shows the past observed trajectory for last $t_p = 5$ seconds in blue. The second column shows the heatmap from $\mathbf{U}_g$ for $t_f = 30$ seconds in the future (goal multimodality) and some sampled goals from the estimated distribution. The third column shows trajectory heatmaps from $\mathbf{U}_t$ conditioned on a sampled goal from column two (path multimodality). The last column shows the predicted trajectories, green indicating the ground-truth trajectories and red our multimodal predictions.

each timestep.

## Non-parametric Distribution Sampling

Given a distribution $P$ of the future frame position as a matrix of probabilities $X$, we aim to sample a two-dimensional point as our estimate for the position of the agent. This is difficult to achieve reliably in practice since the estimated distribution $P$ is noisy during the initial training stages. Hence, taking a naive `argmax` is not robust. Instead, we propose to use the `softargmax` operation [93],

$$\texttt{softargmax}(X) = \left( \sum_i i \frac{\sum_j e^{X_{ij}}}{\sum_{i,j} e^{X_{ij}}}, \sum_j j \frac{\sum_i e^{X_{ij}}}{\sum_{i,j} e^{X_{ij}}} \right)$$

to approximate the most likely position in a robust fashion.

Further details on the sampling process including Test-Time Sampling Trick and Conditional Waypoint Sampling can be found in Supplementary Section 1.1 and 1.2.

### Loss Function

Since the predictions are explicit probability distributions for each timestep, we impose losses directly on the estimated distribution $\hat{P}$ rather than on the drawn coordinate samples. The ground truth future is represented as a Gaussian heatmap $P$ centered at the observed points with a predetermined variance $\sigma_H$. All three networks, $\mathbf{U}_e$, $\mathbf{U}_g$ and $\mathbf{U}_t$ are trained end-to-end jointly using a weighted combination of binary cross entropy losses on the predicted goal, waypoint and trajectory distributions.

$$\mathcal{L}_{\text{goal}} = \text{BCE}(P(\mathbf{u}_{n_p+n_f}), \hat{P}(\mathbf{u}_{n_p+n_f}))$$

$$\mathcal{L}_{\text{waypoint}} = \sum_{i=1}^{N^w} \text{BCE}(P(\mathbf{u}_{w_i}), \hat{P}(\mathbf{u}_{w_i}))$$

$$\mathcal{L}_{\text{trajectory}} = \sum_{i=n_p+1}^{n_p+n_f} \text{BCE}(P(\mathbf{u}_i), \hat{P}(\mathbf{u}_i))$$

$$\mathcal{L} = \mathcal{L}_{\text{goal}} + \lambda_1 \mathcal{L}_{\text{waypoint}} + \lambda_2 \mathcal{L}_{\text{trajectory}}$$

## 3.4 Results

We use a total of three datasets to study Y-net's performance – the Stanford Drone Dataset (SDD) [229], the Intersection Drone Dataset (InD) [24], and the ETH [214] / UCY [161] forecasting benchmark.

### Stanford Drone Dataset

We benchmark our proposed model on the popular Stanford drone dataset [229] where several recently proposed methods have improved state-of-the-art performance significantly in the past few years [53]. The dataset is comprised of more than $11,000$ unique pedestrians across $20$ top-down scenes captured on the Stanford university campus in bird's eye view using a flying drone. For short term prediction, we follow the [233, 196] standard setup and dataset split, sampling at $\texttt{FPS} = 2.5$ yielding a input sequence of length $n_p = 8$ and output of length $n_f = 12$, *i.e.* $t_p = 3.2$ sec, $t_f = 4.8$ sec.

In our proposed long term setting, we sample at $\texttt{FPS} = 1$ thus yielding a $n_p = 5$ for $t_f = 5$ seconds in the past and predicting up to one minute into the future. Further, we label the scenes with semantic segmentation maps consisting of $C = 5$ "stuff" classes, namely [30] pavement, terrain, structure, tree and road, depending on the walking affordability of the surface. We split the dataset's scenes in the same way as the short term setup, to evaluate the performance on unseen scenes during training.

| | Stanford Drone Dataset | | | | | | Intersection Drone Dataset | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | S-GAN | PECNet | R-PECNet | Y-net (Ours) | | | S-GAN | PECNet | R-PECNet | Y-net (Ours) | | |
| $K_a$ | 1 | 1 | 1 | 1 | 2 | 5 | 1 | 1 | 1 | 1 | 2 | 5 |
| ADE | 155.32 | 72.22 | 261.27 | 47.94 | 44.94 | 39.49 | 38.57 | 20.25 | 341.80 | 14.99 | 14.02 | 12.67 |
| FDE | 307.88 | 118.13 | 750.42 | 66.71 | 66.71 | 66.71 | 84.61 | 32.95 | 1702.64 | 21.13 | 21.13 | 21.13 |

Table 3.3: **Long term trajectory forecasting results**: We benchmark performance on our proposed long horizon forecasting setting predicting $t_f = 30$ second into the future given $t_p = 5$ second past motion history. All reported errors are in pixels (lower is better) for $K_e = 20$ with additional results for varying $K_a$ with a fixed $K_e$.

### Intersection Drone Dataset

We propose to use the Intersection drone dataset [24] for benchmarking long term trajectory forecasting. The dataset comprises over $10$ hours of measurements over $4$ distinct intersection in an urban environment. The dataset is recorded in $\texttt{FPS} = 25$. We downsample the trajectories to $\texttt{FPS} = 1$ to match our SDD long term setting, filter out non-pedestrian and short trajectories and use a sliding window approach without overlap to split long trajectories. After the preprocessing steps, inD contains 1,396 long term trajectories with $n_p = 5$ and $n_f = 30$. To evaluate performance on unseen environments, we are using location ID $4$ only during testing time. The scene is labeled with the same $C = 5$ classes as in SDD. We convert the coordinates from world coordinates (meters) into pixel coordinates using the provided scale factors from the authors and evaluate metrics in pixels.

### ETH & UCY datasets

The ETH/UCY benchmarks have been widely used for benchmarking trajectory forecasting models in the short horizon setting in recent years [52]. Forecasting performance has improved by over $\sim 64\%$ on average, within the last two years itself [106]. It comprises of five different scenes all of which report position in world coordinates (in meters). We follow the leave one out validation strategy as outlines in prior work [233, 106, 59]. For all ETH & UCY datasets, since the classes of affordances furnished by the surfaces present is small, we use $C = 2$, identifying each pixel as either belonging to class 'road' or 'not road'. Similar to short term SDD, the frames are sampled at $\texttt{FPS = 2.5}$ predicting $n_f = 12$ frames, $t_f = 4.8$ seconds into the future given the last $n_p = 8$ frames comprising of $t_p = 3.2$ seconds of motion history.

### Implementation Details

We train the entire network end to end with Adam optimizer [146] with a learning rate of $1 \times 10^{-4}$ and batch size of 8. A pre-trained segmentation model is used that is finetuned on the specific dataset. Further details are mentioned in the supplementary materials.

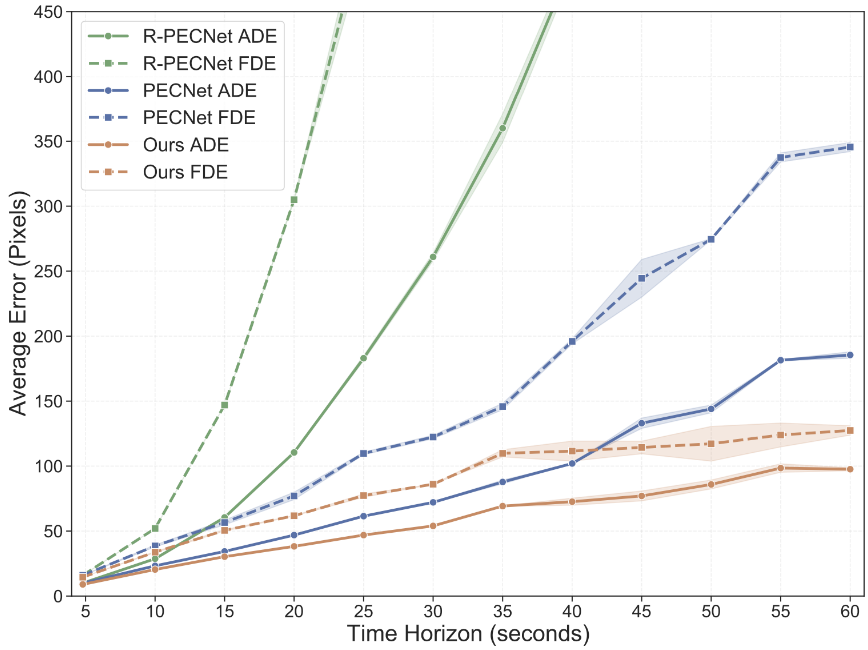



Figure 3.4: **Benchmarking Performance against Time Horizons**: On prediction horizons up to a minute, we observe a consistently growing difference in ADE between Y-net and PECNet, highlighting the importance of factorized goal and path modeling in long term forecasting.

### Metrics

We use the established Average Displacement Error (ADE) and Final Displacement Error (FDE) metrics for measuring performance of future predictions. ADE is calculated as the $\ell_2$ error between the predicted future and the ground truth averaged over the entire trajectory while FDE is the $\ell_2$ error between the predicted future and ground truth for the final predicted point [2]. Following prior works [106], in the case of multiple future predictions, the final error is reported as the `min` error over all predicted futures.

ADE and FDE are well suited metrics for deterministic performance evaluation. However, they use samples instead of the predicted distribution for error estimation. Hence, we report the kernel density estimate-based negative log-likelihood (KDE-NLL) metric in the same fashion as [136, 236]. A standardized KDE is used to estimate a probability distribution function for each predicted future timestep, and the NLL of the ground-truth trajectory is calculated using it. Note, that Y-net predicts explicit probability maps. To be consistent with previous literature and enable fair comparison with baselines we also apply the KDE.

## Short Term Forecasting Results

**Stanford Drone Results**: Table 3.1 presents results for the SDD in the short term setting. We report results with $K_e = 5$ and 20. Since there is limited aleatoric multimodality in short term settings, we use $K_a = 1$ thus being comparable to prior works using 20 trajectory samples for evaluation. Table 3.1 shows our proposed model achieving an ADE of 7.85 and FDE of 11.85 at $K_e = 20$ which outperforms the previous state-of-the-art performance of LB-EBM [209] by 13.0% on ADE and 31.7% on FDE. Further, at $K = 5$ it achieves an ADE of 11.49 and FDE of 20.23 outperforming previous state-of-the-art performance of TNT [319].
**ETH/UCY Results**: We report results on the ETH/UCY benchmark in Table 3.2. Similar to SDD, we set $K_e = 20, K_a = 1$. We observe that Y-net improves the state-of-the-art performance from AgentFormer [311] in FDE by 7.4% to 0.27 and performs on par in ADE with 0.18.

## Long Term Forecasting Results

To study the effect of *epistemic* and *aleatoric* uncertainty factorization, we propose a long term trajectory forecasting setting with a prediction horizon up to 10 times longer than prior works (up to a minute). To benchmark, we retrain PECNet [196], the previous state-of-the-art method from short term forecasting and Social GAN [106] for each prediction horizon setting separately. We also train a recurrent short term baseline based on PECNet (R-PECNet) where the model is trained only for $t_f = 5$ seconds and is fed its own predictions recurrently for predicting longer horizons.

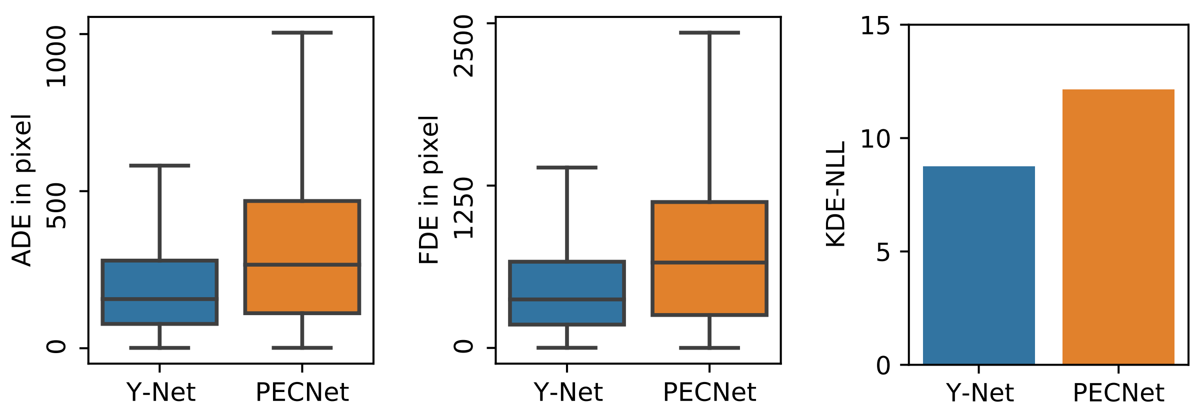



Figure 3.5: **ADE & FDE boxplot and KDE-NLL**: Left and middle: Boxplots of ADE and FDE, respectively. Right: Results for the KDE-NLL metric. All metrics are estimated for the long term setting on SDD with 100 samples.

### Forecasting Results

Table 3.3 reports the baseline and our results on SDD and InD for a time horizon of $t_f = 30$ seconds in the future given the past $t_p = 5$ second input. All reported results are with $K_e = 20$ for Y-net conditioned on $N^w = 1$ intermediate waypoint at $w_1 = 20$, *i.e.* temporally midway between

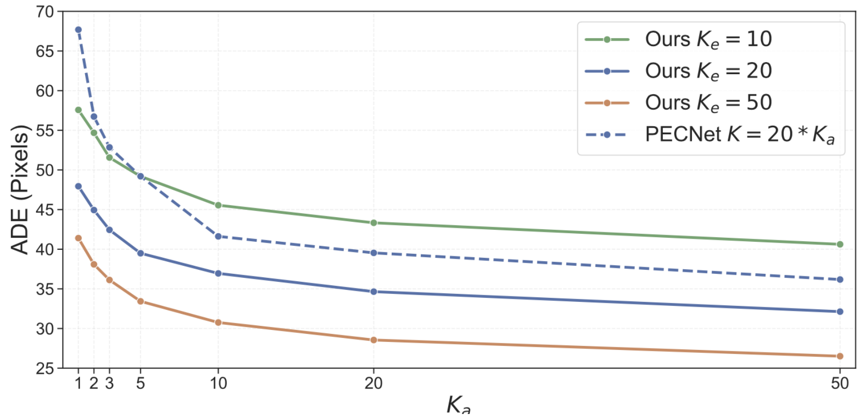



Figure 3.6: **Benchmarking performance against aleatoric uncertainty** ($K_a$): Fixing the goal multimodality ($K_e$) we vary $K_a$ to observe the effect of path multimodality. Also, we benchmark against PECNet by allowing it 20 times more samples for each $K_a$ for a fair compare against the $K_e = 20$ Y-net curve.

the observed inputs and the estimated goal. All reported baseline results are at $K = 20$ for fair comparisons with our $K_e = 20, K_a = 1$ setting. On SDD, we observe that our proposed model outperforms the state-of-the-art short term baseline on the long horizon setting, achieving an ADE of $47.94$ and FDE $66.72$ improving upon PECNet's performance by over $50\%$. Similarly, Y-net outperforms PECNet on InD improving ADE performance from $20.25$ to $14.99$ and FDE from $32.95$ to $21.13$.

To gain a more complete assessment of the performance, boxplots of PECNet and Y-net are shown in Fig. 3.5. These display the median performance and variability of quartiles within the long-term predictions on SDD. Y-net has about half the median error and is much more consistent with less spread.

Further, Y-net achieves a KDE-NLL [136] score of $8.75$, significantly better than PECNet's score of $12.15$ on the same long-term setting on SDD (Fig. 3.5). These additional metrics confirm our observations from the ADE and FDE metrics.

### Varying Prediction Horizon

We compare Y-net with PECnet and R-PECNet for varying prediction horizons. In Fig. 3.4 we observe that the difference in performance between Y-net and PECNet grows as prediction horizon increases from $5$ to $60$ seconds. This shows Y-net's adaptability for long prediction horizons owing to factorized multimodality modeling. We also observe that for PECNet, training a separate model for different time horizons is significantly better than using a short temporal horizon model recurrently (R-PECNet). This motivates our proposal for studying long term forecasting since short term models behave very poorly when applied out of the box recurrently to longer term settings.

### Varying $\mathbf{K_a}$

We also report results with $K_a = 2\ and\ 5$ for studying the improvement in performance from aleatoric multimodality in Table 3.3. We observe a consistent improvement in ADE on both datasets, thus indicating the diversity in predicted paths given the same estimated final goal $\mathbf{u}_{n_p+n_f}$. We also report extensive results for varying the path multimodality $K_a$ with a fixed $K_e$ for various choice of $K_e$ and $K_a$ in Figure 3.6. Additionally for baselining, we benchmark against PECNet [194] evaluated with $K_e$ times more samples than the corresponding Y-net model while varying $K_a$. We show consistent ADE improvements for various $K_e$ when increasing $K_a$, indicating effective use of multimodality. Further, even with $K_e = 20$ times more additional samples, PECNet's performance is significantly worse than Y-net at $K_e = 20$ for all $K_a$ highlighting the importance of factorizing goal and path multimodality for diverse and accurate future trajectory modeling.

### Qualitative Results

We show some qualitative results for long term trajectory prediction ($t_f = 30$) on SDD in Figure 3.3. We observe that Y-net predicts diverse scene-complaint trajectories, with both future goals and paths modalities.

## 3.5 Conclusion

In summary, we present Y-net, a scene-compliant trajectory forecasting network with factorized goal and path multimodalities. Y-net uses the U-net structure [230] for explicitly modeling probability heatmaps for epistemic and aleatoric uncertainties. Overall, Y-net decrease the error of previous state-of-the-art performance by up to $31.7\%$ on the SDD and by up to $7.4\%$ on ETH/UCY benchmarks in the short term setting. We also propose a new long term trajectory forecasting setting with a prediction horizon of up to a minute for exemplifying the epistemic and aleatoric uncertainty dichotomy. In this setting, we benchmark on the Stanford Drone and Intersection Drone dataset where Y-net exceeds previous state-of-the-art by over $77.1\%$ and $56.0\%$ respectively thereby highlighting the importance of modeling factorized stochasticity.

# Chapter 4

# Multiscale Vision Transformers

## 4.1 Introduction

We begin with the intellectual history of neural network models for computer vision. Based on their studies of cat and monkey visual cortex, Hubel and Wiesel [133] developed a *hierarchical* model of the visual pathway with neurons in lower areas such as V1 responding to features such as oriented edges and bars, and in higher areas to more specific stimuli. Fukushima proposed the Neocognitron [81], a neural network architecture for pattern recognition explicitly motivated by Hubel and Wiesel's hierarchy. His model had alternating layers of simple cells and complex cells, thus incorporating downsampling, and shift invariance, thus incorporating convolutional structure. LeCun *et al*. [157] took the additional step of using backpropagation to train the weights of this network. But already the main aspects of hierarchy of visual processing had been established: (*i*) Reduction in spatial resolution as one goes up the processing hierarchy and (*ii*) Increase in the number of different "channels", with each channel corresponding to ever more specialized features.

In a parallel development, the computer vision community developed *multiscale* processing, sometimes called "pyramid" strategies, with Rosenfeld and Thurston [231], Burt and Adelson [29], Koenderink [151], among the key papers. There were two motivations (*i*) To decrease the computing requirements by working at lower resolutions and (*ii*) A better sense of "context" at the lower resolutions, which could then guide the processing at higher resolutions (this is a precursor to the benefit of "depth" in today's neural networks.)

The Transformer [270] architecture allows learning arbitrary functions defined over sets and has been scalably successful in sequence tasks such as language comprehension [69] and machine translation [27]. Fundamentally, a transformer uses blocks with two basic operations. First, is an attention operation [13] for modeling inter-element relations. Second, is a multi-layer perceptron

This chapter is based on joint work with Haoqi Fan, Bo Xiong, Yanghao Li, Zhicheng Yan, Jitendra Malik and Christoph Feichtenhofer [71], and is presented as it appeared in the ICCV 2021 proceedings.

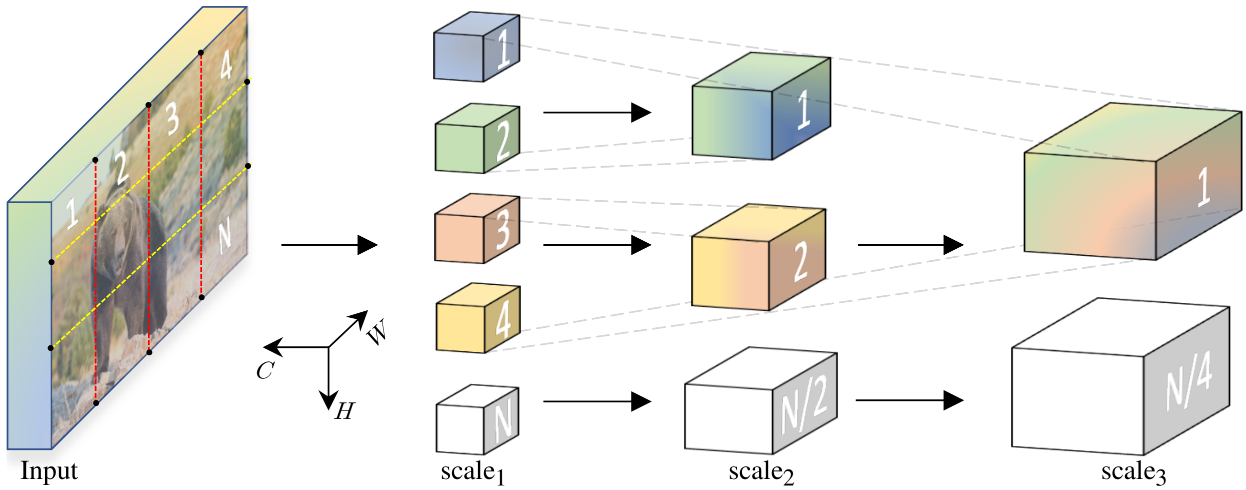



Figure 4.1: **Multiscale Vision Transformers** learn a hierarchy from *dense* (in space) and *simple* (in channels) to *coarse* and *complex* features. Several resolution-channel *scale* stages progressively *increase* the channel capacity of the intermediate latent sequence while *reducing* its length and thereby spatial resolution.

(MLP), which models relations within an element. Intertwining these operations with normalization [11] and residual connections [112] allows transformers to generalize to a wide variety of tasks.

Recently, transformers have been applied to key computer vision tasks such as image classification. In the spirit of architectural universalism, vision transformers [67, 264] approach performance of convolutional models across a variety of data and compute regimes. By only having a first layer that 'patchifies' the input in spirit of a 2D convolution, followed by a stack of transformer blocks, the vision transformer aims to showcase the power of the transformer architecture using little inductive bias.

In this paper, our intention is to connect the seminal idea of *multiscale feature hierarchies* with the transformer model. We posit that the fundamental vision principle of resolution and channel scaling, can be beneficial for transformer models across a variety of visual recognition tasks.

We present Multiscale Vision Transformers (MViT), a transformer architecture for modeling visual data such as images and videos. Consider an input image as shown in Fig. 5.1. Unlike conventional transformers, which maintain a constant channel capacity and resolution throughout the network, Multiscale Transformers have several channel-resolution 'scale' stages. Starting from the image resolution and a small channel dimension, the stages *hierarchically expand* the *channel* capacity while *reducing* the *spatial* resolution. This creates a multiscale pyramid of feature activations inside the transformer network, effectively connecting the principles of transformers with multi scale feature hierarchies.

Our conceptual idea provides an effective design advantage for vision transformer models. The early layers of our architecture can operate at high spatial resolution to model *simple* low-level visual information, due to the lightweight channel capacity. In turn, the deeper layers can effectively focus

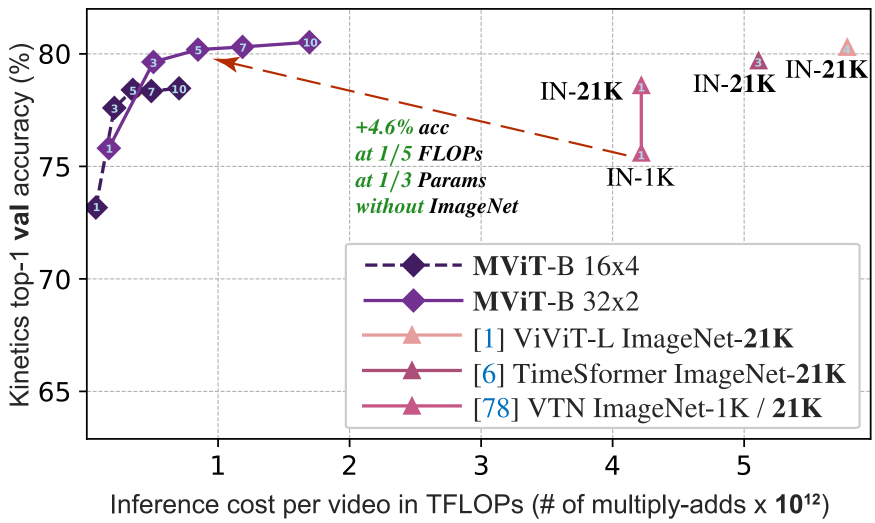



Figure 4.2: **Accuracy/complexity trade-off** on Kinetics-400 for varying # of inference clips per video shown in MViT curves. Concurrent vision-transformer based methods [204, 22, 8] require over 5× more computation *and large-scale external pre-training* on ImageNet-**21K** (IN-**21K**), to achieve equivalent **MViT** accuracy.

on spatially coarse but *complex* high-level features to model visual semantics. The fundamental advantage of our multiscale transformer arises from the extremely dense nature of visual signals, a phenomenon that is even more pronounced for space-time visual signals captured in *video*.

A noteworthy benefit of our design is the presence of strong implicit temporal bias in video multiscale models. We show that vision transformer models [67] trained on natural video suffer no performance decay when tested on videos with *shuffled* frames. This indicates that these models are not effectively using the temporal information and instead rely heavily on appearance. In contrast, when testing our MViT models on shuffled frames, we observe significant accuracy decay, indicating strong use of temporal information.

Our focus in this paper is video recognition, and we design and evaluate MViT for video tasks (Kinetics [143, 36], Charades [241], SSv2 [97] and AVA [103]). MViT provides a significant performance gain over concurrent video transformers [204, 22, 8], *without* any external pre-training data.

In Fig. A.1 we show the computation/accuracy trade-off for video-level inference, when varying the number of temporal clips used in MViT. The vertical axis shows accuracy on Kinetics-400 and the horizontal axis the overall inference cost in FLOPs for different models, **MViT** and concurrent ViT [67] video variants: VTN [204], TimeSformer [22], ViViT [8]. To achieve similar accuracy level as MViT, these models require significant more computation and parameters (*e.g*. ViViT-L [8] has 6.8× higher FLOPs and 8.5× more parameters at equal accuracy, more analysis in §A.2) and need large-scale external pre-training on ImageNet-**21K** (which contains around 60× more labels than Kinetics-400).

We further apply MViT to ImageNet [58] classification, by *simply removing the temporal dimension* of the video architecture, and show significant gains over single-scale vision transformers for image recognition. Our code and models are available in PySlowFast [72] and PyTorchVideo [73].

## 4.2 Related Work

**Convolutional networks (ConvNets)**. Incorporating downsampling, shift invariance, and shared weights, ConvNets are de-facto standard backbones for computer vision tasks for image [157, 152, 243, 253, 113, 42, 46, 84, 255, 221, 108] and video [242, 76, 34, 219, 172, 293, 266, 77, 287, 88, 75, 326].

**Self-attention in ConvNets.** Self-attention mechanisms has been used for image understanding [222, 320, 126, 19], unsupervised object recognition [188] as well as vision and language [191, 166]. Hybrids of self-attention operations and convolutional networks have also been applied to image understanding [127] and video recognition [277].

**Vision Transformers.** Much of current enthusiasm in application of Transformers [270] to vision tasks commences with the Vision Transformer (ViT) [67] and Detection Transformer [33]. We build directly upon [67] with a staged model allowing channel expansion and resolution downsampling. DeiT [264] proposes a data efficient approach to training ViT. Our training recipe builds on, and we compare our image classification models to, DeiT under identical settings.

An emerging thread of work aims at applying transformers to vision tasks such as object detection [16], semantic segmentation [321, 273], 3D reconstruction [179], pose estimation [304], generative modeling [45], image retrieval [205], medical image segmentation [43, 268, 313], point clouds [105], video instance segmentation [280], object re-identification [116], video retrieval [82], video dialogue [156], video object detection [312] and multi-modal tasks [184, 61, 220, 128, 308]. A separate line of works attempts at modeling visual data with learnt discretized token sequences [284, 223, 45, 310, 51].

**Efficient Transformers.** Recent works [274, 149, 50, 258, 55, 48, 168, 18] reduce the quadratic attention complexity to make transformers more efficient for natural language processing applications, which is complementary to our approach.

Three concurrent works propose a ViT-based architecture for video [204, 22, 8]. However, these methods rely on pre-training on vast amount of external data such as ImageNet-21K [58], and thus use the vanilla ViT [67] with minimal adaptations. In contrast, our MViT introduces multiscale feature hierarchies for transformers, allowing effective modeling of dense visual input without large-scale external data.

## 4.3 Multiscale Vision Transformer (MViT)

Our generic Multiscale Transformer architecture builds on the core concept of *stages*. Each stage consists of multiple transformer blocks with specific space-time resolution and channel dimension.

The main idea of Multiscale Transformers is to progressively *expand* the channel capacity, while *pooling* the resolution from input to output of the network.

## Multi Head Pooling Attention

We first describe Multi Head Pooling Attention (MHPA), a self attention operator that enables flexible resolution modeling in a transformer block allowing Multiscale Transformers to operate at progressively changing spatiotemporal resolution. In contrast to original Multi Head Attention (MHA) operators [270], where the channel dimension and the spatiotemporal resolution remains fixed, MHPA *pools* the sequence of latent tensors to reduce the sequence length (resolution) of the attended input. Fig. 4.3 shows the concept.

Concretely, consider a $D$ dimensional input tensor $X$ of sequence length $L$, $X \in \mathbb{R}^{L\times D}$. Following MHA [67], MHPA projects the input $X$ into intermediate query tensor $\hat{Q} \in \mathbb{R}^{L\times D}$, key tensor $\hat{K} \in \mathbb{R}^{L\times D}$ and value tensor $\hat{V} \in \mathbb{R}^{L\times D}$ with linear operations

$$\hat{Q} = XW_Q \quad \hat{K} = XW_K \quad \hat{V} = XW_V$$

/ with weights $W_Q, W_K, W_V$ of dimensions $D \times D$. The obtained intermediate tensors are then pooled in sequence length, with a pooling operator $\mathcal{P}$ as described below.

**Pooling Operator.** Before attending the input, the intermediate tensors $\hat{Q}, \hat{K}, \hat{V}$ are pooled with the pooling operator $\mathcal{P}(\cdot; \mathbf{\Theta})$ which is the cornerstone of our MHPA and, by extension, of our Multiscale Transformer architecture.

The operator $\mathcal{P}(\cdot; \mathbf{\Theta})$ performs a pooling kernel computation on the input tensor along each of the dimensions. Unpacking $\mathbf{\Theta}$ as $\mathbf{\Theta} := (\mathbf{k}, \mathbf{s}, \mathbf{p})$, the operator employs a pooling kernel $\mathbf{k}$ of dimensions $k_T \times k_H \times k_W$, a stride $\mathbf{s}$ of corresponding dimensions $s_T \times s_H \times s_W$ and a padding $\mathbf{p}$ of corresponding dimensions $p_T \times p_H \times p_W$ to reduce an input tensor of dimensions $\mathbf{L} = T \times H \times W$ to $\tilde{\mathbf{L}}$ given by,

$$\tilde{\mathbf{L}} = \left\lfloor \frac{\mathbf{L} + 2\mathbf{p} - \mathbf{k}}{\mathbf{s}} \right\rfloor + 1$$

with the equation applying coordinate-wise. The pooled tensor is flattened again yielding the output of $\mathcal{P}(Y; \mathbf{\Theta}) \in \mathbb{R}^{\tilde{L}\times D}$ with reduced sequence length, $\tilde{L} = \tilde{T} \times \tilde{H} \times \tilde{W}$.

By default we use *overlapping* kernels $\mathbf{k}$ with *shape-preserving* padding $\mathbf{p}$ in our pooling attention operators, so that $\tilde{L}$ , the sequence length of the output tensor $\mathcal{P}(Y; \mathbf{\Theta})$, experiences an overall reduction by a factor of $s_T s_H s_W$.

**Pooling Attention.** The pooling operator $P(\cdot; \Theta)$ is applied to all the intermediate tensors $\hat{Q}$, $\hat{K}$ and $\hat{V}$ independently with chosen pooling kernels $\mathbf{k}$, stride $\mathbf{s}$ and padding $\mathbf{p}$. Denoting $\theta$ yielding the pre-attention vectors $Q = \mathcal{P}(\hat{Q}; \mathbf{\Theta}_Q)$, $K = \mathcal{P}(\hat{K}; \mathbf{\Theta}_K)$ and $V = \mathcal{P}(\hat{V}; \mathbf{\Theta}_V)$ with reduced sequence lengths. Attention is now computed on these shortened vectors, with the operation,

$$\text{Attention}(Q, K, V) = \text{Softmax}(QK^T/\sqrt{D})V.$$

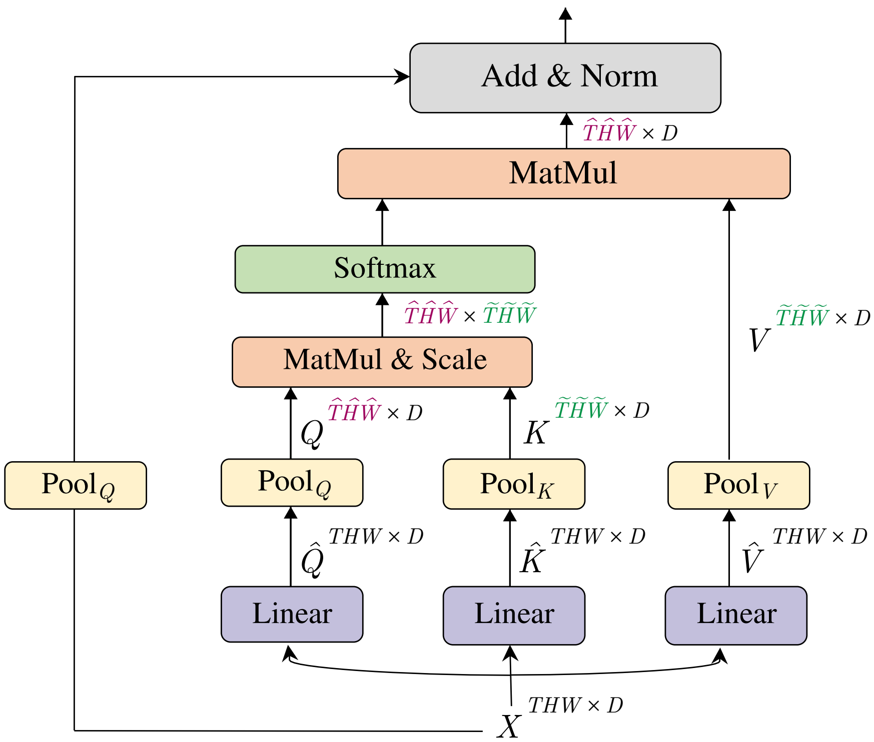



Figure 4.3: **Pooling Attention** is a flexible attention mechanism that (i) allows obtaining the reduced space-time resolution ($\hat{T}\hat{H}\hat{W}$) of the input ($THW$) by pooling the query, $Q = \mathcal{P}(\hat{Q}; \mathbf{\Theta}_Q)$, and/or (ii) computes attention on a reduced length ($\tilde{T}\tilde{H}\tilde{W}$) by pooling the key, $K = \mathcal{P}(\hat{K}; \mathbf{\Theta}_K)$, and value, $V = \mathcal{P}(\hat{V}; \mathbf{\Theta}_V)$, sequences.

Naturally, the operation induces the constraints $\mathbf{s}_K \equiv \mathbf{s}_V$ on the pooling operators. In summary, pooling attention is computed as,

$$\mathrm{PA}(\cdot) = \mathrm{Softmax}(\mathcal{P}(Q; \mathbf{\Theta}_Q)\mathcal{P}(K; \mathbf{\Theta}_K)^T/\sqrt{d})\mathcal{P}(V; \mathbf{\Theta}_V),$$

where $\sqrt{d}$ is normalizing the inner product matrix row-wise. The output of the Pooling attention operation thus has its sequence length reduced by a *stride* factor of $s_T^Q s_H^Q s_W^Q$ following the shortening of the query vector $Q$ in $\mathcal{P}(\cdot)$.

**Multiple heads.** As in [270] the computation can be parallelized by considering $h$ heads where each head is performing the pooling attention on a non overlapping subset of $D/h$ channels of the $D$ dimensional input tensor $X$.

**Computational Analysis.** Since attention computation scales quadratically w.r.t. the sequence length, pooling the key, query and value tensors has dramatic benefits on the fundamental compute and memory requirements of the Multiscale Transformer model. Denoting the sequence length reduction factors by $f_Q$, $f_K$ and $f_V$ we have,

$$f_j = s_T^j \cdot s_H^j \cdot s_W^j, \; \forall \; j \in \{Q, K, V\}.$$

| stage | operators | output sizes |
|---|---|---|
| data layer | stride $\tau \times 1 \times 1$ | $T \times H \times W$ |
| $\text{patch}_1$ | $1\times16\times16$, D<br>stride $1\times16\times16$ | $D\times T\times\frac{H}{16}\times\frac{W}{16}$ |
| $\text{scale}_2$ | $\begin{bmatrix}\text{MHA}(D)\\ \text{MLP}(4D)\end{bmatrix}\times N$ | $D\times T\times\frac{H}{16}\times\frac{W}{16}$ |

Table 4.1: **Vision Transformers (ViT)** base model starts from a data layer that samples visual input with rate $\tau\times1\times1$ to $T\times H\times W$ resolution, where $T$ is the number of frames $H$ height and $W$ width. The first layer, $\text{patch}_1$ projects patches (of shape $1\times16\times16$) to form a sequence, processed by a stack of $N$ transformer blocks ($\text{stage}_2$) at uniform channel dimension ($D$) and resolution ($T\times\frac{H}{16}\times\frac{W}{16}$).

Considering the input tensor to $\mathcal{P}(;\Theta)$ to have dimensions $D \times T \times H \times W$, the run-time complexity of MHPA is $O(THWD/h(D + THW/f_Q f_K))$ per head and the memory complexity is $O(THWh(D/h + THW/f_Q f_K))$.

This trade-off between the number of channels $D$ and sequence length term $THW/f_Q f_K$ informs our design choices about architectural parameters such as number of heads and width of layers. We refer the reader to §A.4 for a detailed analysis and discussions on the runtime-memory complexity trade-off.

## Multiscale Transformer Networks

Building upon Multi Head Pooling Attention (Sec. 4.3), we describe the Multiscale Transformer model for visual representation learning using exclusively MHPA and MLP layers. First, we present a brief review of the Vision Transformer Model that informs our design.

**Preliminaries: Vision Transformer (ViT).** The Vision Transformer (ViT) architecture [67] starts by dicing the input video of resolution $T\times H\times W$, where $T$ is the number of frames $H$ the height and $W$ the width, into non-overlapping patches of size $1\times16\times16$ each, followed by point-wise application of linear layer on the flattened image patches to to project them into the latent dimension, $D$, of the transformer. This is *equivalent* to a convolution with equal kernel size and stride of $1\times16\times16$ and is shown as $\text{patch}_1$ stage in the model definition in Table 4.1.

Next, a positional embedding $\mathbf{E} \in \mathbb{R}^{L\times D}$ is added to each element of the projected sequence of length $L$ with dimension $D$ to encode the positional information and break permutation invariance. A learnable class embedding is appended to the projected image patches.

The resulting sequence of length of $L + 1$ is then processed sequentially by a stack of $N$ transformer blocks, each one performing attention ($\text{MHA}$ [270]), multi-layer perceptron ($\text{MLP}$) and layer normalization ($\text{LN}$[12]) operations. Considering $X$ to be the input of the block, the output of a single transformer block, $\text{Block}(X)$ is computed by

$$X_1 = \text{MHA}(\text{LN}(X)) + X$$
$$\text{Block}(X) = \text{MLP}(\text{LN}(X_1)) + X_1.$$

| stages | operators | output sizes |
|---|---|---|
| data layer | stride $\tau{\times}1{\times}1$ | $D{\times}T{\times}H{\times}W$ |
| cube$_1$ | $c_T{\times}c_H{\times}c_W$, $D$<br>stride $s_T{\times}4{\times}4$ | $D{\times}\frac{T}{s_T}{\times}\frac{H}{4}{\times}\frac{W}{4}$ |
| scale$_2$ | $\begin{bmatrix}\text{MHPA}(D)\\ \text{MLP}(4D)\end{bmatrix}{\times}N_2$ | $D{\times}\frac{T}{s_T}{\times}\frac{H}{4}{\times}\frac{W}{4}$ |
| scale$_3$ | $\begin{bmatrix}\text{MHPA}(2D)\\ \text{MLP}(8D)\end{bmatrix}{\times}N_3$ | $2D{\times}\frac{T}{s_T}{\times}\frac{H}{8}{\times}\frac{W}{8}$ |
| scale$_4$ | $\begin{bmatrix}\text{MHPA}(4D)\\ \text{MLP}(16D)\end{bmatrix}{\times}N_4$ | $4D{\times}\frac{T}{s_T}{\times}\frac{H}{16}{\times}\frac{W}{16}$ |
| scale$_5$ | $\begin{bmatrix}\text{MHPA}(8D)\\ \text{MLP}(32D)\end{bmatrix}{\times}N_5$ | $8D{\times}\frac{T}{s_T}{\times}\frac{H}{32}{\times}\frac{W}{32}$ |

Table 4.2: **Multiscale Vision Transformers (MViT)** base model. Layer cube$_1$, projects *dense* space-time cubes (of shape $c_t{\times}c_y{\times}c_w$) to $D$ channels to reduce spatiotemporal resolution to $\frac{T}{s_T}{\times}\frac{H}{4}{\times}\frac{W}{4}$. The subsequent stages progressively down-sample this resolution (at beginning of a stage) with MHPA while simultaneously increasing the channel dimension, in MLP layers, (at the end of a stage). Each stage consists of $N_*$ transformer blocks, denoted in [brackets].

The resulting sequence after $N$ consecutive blocks is layer-normalized and the class embedding is extracted and passed through a linear layer to predict the desired output (*e.g*. class). By default, the hidden dimension of the MLP is $4D$. We refer the reader to[67, 270] for details.

In context of the present paper, it is noteworthy that ViT maintains a constant channel capacity and spatial resolution throughout all the blocks (see Table 4.1).

**Multiscale Vision Transformers (MViT).** Our key concept is to progressively *grow* the *channel* resolution (*i.e*. dimension), while simultaneously *reducing* the *spatiotemporal* resolution (*i.e*. sequence length) throughout the network. By design, our MViT architecture has *fine* spacetime (and *coarse* channel) resolution in early layers that is up-/downsampled to a coarse spacetime (and *fine* channel) resolution in late layers. MViT is shown in Table 4.2.

**Scale stages.** A *scale stage* is defined as a set of $N$ transformer blocks that operate on the same *scale* with identical resolution across channels and space-time dimensions $D{\times}T{\times}H{\times}W$. At the input (cube$_1$ in Table 4.2), we project the patches (or cubes if they have a temporal extent) to a smaller channel dimension (*e.g*. $8\times$ smaller than a typical ViT model), but long sequence (*e.g*. $4{\times}4 = 16\times$ denser than a typical ViT model; *cf.* Table 4.1).

At a stage *transition* (*e.g*. scale$_1$ to scale$_2$ to in Table 4.2), the channel dimension of the processed sequence is up-sampled while the length of the sequence is down-sampled. This effectively reduces the spatiotemporal resolution of the underlying visual data while allowing the network to assimilate the processed information in more complex features.

**Channel expansion.** When transitioning from one stage to the next, we expand the channel dimension by increasing the output of the final MLP layer in the previous stage by a factor that is relative to the resolution change introduced at the stage. Concretely, if we down-sample the

| stage | operators | | output sizes |
|---|---|---|---|
| data | stride $8\times1\times1$ | | $8\times224\times224$ |
| patch$_1$ | $1\times16\times16$, 768<br>stride $1\times16\times16$ | | $768\times8\times14\times14$ |
| scale$_2$ | MHA(768)<br>MLP(3072) | $\times12$ | $768\times8\times14\times14$ |

(a) **ViT**-B with **179.6**G FLOPs, **87.2**M param, **16.8**G memory, and **68.5**% top-1 accuracy.

| stage | operators | | output sizes |
|---|---|---|---|
| data | stride $4\times1\times1$ | | $16\times224\times224$ |
| cube$_1$ | $3\times7\times7$, 96<br>stride $2\times4\times4$ | | $96\times8\times56\times56$ |
| scale$_2$ | MHPA(96)<br>MLP(384) | $\times1$ | $96\times8\times56\times56$ |
| scale$_3$ | MHPA(192)<br>MLP(768) | $\times2$ | $192\times8\times28\times28$ |
| scale$_4$ | MHPA(384)<br>MLP(1536) | $\times11$ | $384\times8\times14\times14$ |
| scale$_5$ | MHPA(768)<br>MLP(3072) | $\times2$ | $768\times8\times7\times7$ |

(b) **MViT**-B with **70.5**G FLOPs, **36.5**M param, **6.8**G memory, and **77.2**% top-1 accuracy.

| stage | operators | | output sizes |
|---|---|---|---|
| data | stride $4\times1\times1$ | | $16\times224\times224$ |
| cube$_1$ | $3\times8\times8$, 128<br>stride $2\times8\times8$ | | $128\times8\times28\times28$ |
| scale$_2$ | MHPA(128)<br>MLP(512) | $\times3$ | $128\times8\times28\times28$ |
| scale$_3$ | MHPA(256)<br>MLP(1024) | $\times7$ | $256\times8\times14\times14$ |
| scale$_4$ | MHPA(512)<br>MLP(2048) | $\times6$ | $512\times8\times7\times7$ |

(c) **MViT**-S with **32.9**G FLOPs, **26.1**M param, **4.3**G memory, and **74.3**% top-1 accuracy.

Table 4.3: Comparing **ViT**-B to two instantiations of **MViT** with varying complexity, **MViT**-S in (c) and **MViT**-B in (b). **MViT**-S operates at a lower spatial resolution and lacks a first high-resolution stage. The top-1 accuracy corresponds to *5-Center* view testing on K400. FLOPs correspond to a single inference clip, and memory is for a training batch of 4 clips. See Table 4.2 for the general **MViT**-B structure.

space-time resolution by $4\times$, we increase the channel dimension by $2\times$. For example, scale$_3$ to scale$_4$ changes resolution from $2D\times\frac{T}{s_T}\times\frac{H}{8}\times\frac{T}{8}$ to $4D\times\frac{T}{s_T}\times\frac{H}{16}\times\frac{T}{16}$ in Table 4.2. This roughly preserves the computational complexity across stages, and is similar to ConvNet design principles [242, 111].

**Query pooling.** The pooling attention operation affords flexibility not only in the length of key and value vectors but also in the length of the query, and thereby output, sequence. Pooling the query vector $\mathcal{P}(Q;\mathbf{k};\mathbf{p};\mathbf{s})$ with a kernel $\mathbf{s}\equiv(s^Q_T, s^Q_H, s^Q_W)$ leads to sequence reduction by a factor of $s^Q_T\cdot s^Q_H\cdot s^Q_W$. Since, our intention is to decrease resolution at the beginning of a stage and then preserve this resolution throughout the stage, only the first pooling attention operator of each stage operates at non-degenerate query stride $\mathbf{s}^Q>1$, with all other operators constrained to

$\mathbf{s}^Q \equiv (1, 1, 1)$.

**Key-Value pooling.** Unlike Query pooling, changing the sequence length of key $K$ and value $V$ tensors, does not change the output sequence length and, hence, the space-time resolution. However, they play a key role in overall computational requirements of the pooling attention operator.

We decouple the usage of $K, V$ and $Q$ pooling, with $Q$ pooling being used in the first layer of each stage and $K, V$ pooling being employed in all other layers. Since the sequence length of key and value tensors need to be identical to allow attention weight calculation, the pooling stride used on $K$ and value $V$ tensors needs to be identical. In our default setting, we constrain *all* pooling parameters ($\mathbf{k}$; $\mathbf{p}$; $\mathbf{s}$) to be identical *i.e.* $\Theta_K \equiv \Theta_V$ within a stage, but vary $\mathbf{s}$ *adaptively* w.r.t. to the scale across stages.

**Skip connections.** Since the channel dimension and sequence length change inside a residual block, we pool the skip connection to adapt to the dimension mismatch between its two ends. MHPA handles this mismatch by adding the query pooling operator $\mathcal{P}(\cdot; \mathbf{\Theta}_Q)$ to the residual path. As shown in Fig. 4.3, instead of directly adding the input $X$ of MHPA to the output, we add the pooled input $X$ to the output, thereby matching the resolution to attended query $Q$.

For handling the channel dimension mismatch between stage changes, we employ an extra linear layer that operates on the layer-normalized output of our MHPA operation. Note that this differs from the other (resolution-preserving) skip-connections that operate on the un-normalized signal.

## Network instantiation details

Table A.1 shows concrete instantiations of the base models for Vision Transformers [67] and our Multiscale Vision Transformers. ViT-Base [67] (Table A.1b) initially projects the input to patches of shape 1×16×16 with dimension $D = 768$, followed by stacking $N = 12$ transformer blocks. With an 8×224×224 input the resolution is fixed to 768×8×14×14 throughout *all* layers. The sequence length (spacetime resolution + class token) is $8 \cdot 14 \cdot 14 + 1 = 1569$.

Our MViT-Base (Table A.1b) is comprised of $4$ scale stages, each having several transformer blocks of consistent channel dimension. **MViT**-B initially projects the input to a channel dimension of $D = 96$ with *overlapping* space-time cubes of shape 3×7×7. The resulting sequence of length $8 * 56 * 56 + 1 = 25089$ is reduced by a factor of $4$ for each additional stage, to a final sequence length of $8 * 7 * 7 + 1 = 393$ at scale$_4$. In tandem, the channel dimension is up-sampled by a factor of 2 at each stage, increasing to 768 at scale$_4$. Note that all pooling operations, and hence the resolution down-sampling, is performed only on the data sequence without involving the processed class token embedding.

We set the number of MHPA heads to $h = 1$ in the scale$_1$ stage and increase the number of heads with the channel dimension (channels per-head $D/h$ remain consistent at 96).

At each stage transition, the previous stage output MLP dimension is increased by 2× and MHPA pools on $Q$ tensors with $\mathbf{s}^Q = (1, 2, 2)$ at the input of the next stage.

We employ $K, V$ pooling in all MHPA blocks, with $\Theta_K \equiv \Theta_V$ and $\mathbf{s}^{K,V} = (1, 8, 8)$ in scale$_1$ and *adaptively* decay this stride w.r.t. to the scale across stages such that the $K, V$ tensors have consistent scale across all blocks.

## 4.4 Experiments: Video Recognition

**Datasets.** We use Kinetics-400 [143] (K400) (~240k training videos in 400 classes) and Kinetics-600 [36]. We further assess transfer learning performance for on Something-Something-v2 [97], Charades [241], and AVA [103].

We report top-1 and top-5 classification accuracy (%) on the validation set, computational cost (in FLOPs) of a single, spatially center-cropped clip and the number of clips used.

**Training.** By default, all models are trained *from random initialization* ("*from scratch*") on Kinetics, *without* using ImageNet [57] or other pre-training. Our training recipe and augmentations follow [77, 264]. For Kinetics, we train for 200 epochs with 2 repeated augmentation [124] repetitions.

We report ViT baselines that are *fine-tuned* from ImageNet, using a 30-epoch version of the training recipe in [77].

For the temporal domain, we sample a clip from the full-length video, and the input to the network are $T$ frames with a temporal stride of $\tau$; denoted as $T \times \tau$ [77].

**Inference.** We apply two testing strategies following [77, 75]: *(i)* Temporally, uniformly samples $K$ clips (*e.g.* $K$=5) from a video, scales the shorter spatial side to 256 pixels and takes a 224×224 center crop, and *(ii)*, the same as *(i)* temporally, but take 3 crops of 224×224 to cover the longer spatial axis. We average the scores for all individual predictions.

All implementation specifics are in §A.5.

### Main Results

**Kinetics-400.** Table 5.2 compares to prior work. From top-to-bottom, it has four sections and we discuss them in turn.

The first Table 5.2 section shows prior art using ConvNets.

The second section shows concurrent work using Vision Transformers [67] for video classification [204, 22]. Both approaches rely on ImageNet pre-trained base models. ViT-B-VTN [204] achieves 75.6% top-1 accuracy, which is boosted by 3% to 78.6% by merely changing the pre-training from ImageNet-1K to ImageNet-21K. ViT-B-TimeSformer [22] shows another 2.1% gain on top of VTN, at higher cost of 7140G FLOPs and 121.4M parameters. ViViT improves accuracy further with an even larger ViT-L model.

The third section in Table 5.2 shows our ViT baselines. We first list our ViT-B, also pre-trained on the ImageNet-21K, which achieves 79.3%, thereby being 1.4% lower than ViT-B-TimeSformer, but is with 4.4× fewer FLOPs and 1.4× fewer parameters. This result shows that *simply fine-tuning an off-the-shelf ViT-B model from ImageNet-**21K*** [67] provides a strong baseline on Kinetics. However, training this model from-scratch with the same fine-tuning recipe will result in 34.3%. Using our "training-from-scratch" recipe will produce 68.5% for this ViT-B model, using the same 1×5, spatial × temporal, views for video-level inference.

The final section of Table 5.2 lists our **MViT** results. All our models are *trained-from-scratch* using this recipe, *without* any external pre-training. Our small model, **MViT**-S produces 76.0%

| model | pre-train | top-1 | top-5 | FLOPs×views | Param |
|---|---|---|---|---|---|
| Two-Stream I3D [34] | - | 71.6 | 90.0 | 216 × NA | 25.0 |
| ip-CSN-152 [266] | - | 77.8 | 92.8 | 109×3×10 | 32.8 |
| SlowFast 8×8 +NL [77] | - | 78.7 | 93.5 | 116×3×10 | 59.9 |
| SlowFast 16×8 +NL [77] | - | 79.8 | 93.9 | 234×3×10 | 59.9 |
| X3D-M [75] | - | 76.0 | 92.3 | 6.2×3×10 | 3.8 |
| X3D-XL [75] | - | 79.1 | 93.9 | 48.4×3×10 | 11.0 |
| ViT-B-VTN [204] | ImageNet-1K | 75.6 | 92.4 | 4218×1×1 | 114.0 |
| ViT-B-VTN [204] | ImageNet-**21K** | 78.6 | 93.7 | 4218×1×1 | 114.0 |
| ViT-B-TimeSformer [22] | ImageNet-**21K** | 80.7 | 94.7 | 2380×3×1 | 121.4 |
| ViT-L-ViViT [8] | ImageNet-**21K** | 81.3 | 94.7 | 3992×3×4 | 310.8 |
| ViT-B (our baseline) | ImageNet-**21K** | 79.3 | 93.9 | 180×1×5 | 87.2 |
| ViT-B (our baseline) | - | 68.5 | 86.9 | 180×1×5 | 87.2 |
| **MViT**-S | - | 76.0 | 92.1 | 32.9×1×5 | 26.1 |
| **MViT**-B, 16×4 | - | 78.4 | 93.5 | 70.5×1×5 | 36.6 |
| **MViT**-B, 32×3 | - | 80.2 | 94.4 | 170×1×5 | 36.6 |
| **MViT**-B, 64×3 | - | **81.2** | **95.1** | 455×3×3 | 36.6 |

Table 4.4: **Comparison with previous work on Kinetics-400**. We report the inference cost with a single "view" (temporal clip with spatial crop) × the number of views (FLOPs×$\text{view}_{\text{space}}$×$\text{view}_{\text{time}}$). Magnitudes are Giga ($10^9$) for FLOPs and Mega ($10^6$) for Param. Accuracy of models trained with external data is de-emphasized.

| model | pretrain | top-1 | top-5 | GFLOPs×views | Param |
|---|---|---|---|---|---|
| SlowFast 16×8 +NL [77] | - | 81.8 | 95.1 | 234×3×10 | 59.9 |
| X3D-M | - | 78.8 | 94.5 | 6.2×3×10 | 3.8 |
| X3D-XL | - | 81.9 | 95.5 | 48.4×3×10 | 11.0 |
| ViT-B-TimeSformer [22] | IN-**21K** | 82.4 | 96.0 | 1703×3×1 | 121.4 |
| ViT-L-ViViT [8] | IN-**21K** | 83.0 | 95.7 | 3992×3×4 | 310.8 |
| **MViT**-B, 16×4 | - | 82.1 | 95.7 | 70.5×1×5 | 36.8 |
| **MViT**-B, 32×3 | - | 83.4 | 96.3 | 170×1×5 | 36.8 |
| **MViT**-B-24, 32×3 | - | **84.1** | **96.5** | 236×1×5 | 52.9 |

Table 4.5: **Comparison with previous work on Kinetics-600**.

while being relatively lightweight with 26.1M param and 32.9×5=164.5G FLOPs, outperforming ViT-B by **+7.5**% at **5.5**× less compute in *identical* train/val setting.

Our base model, **MViT**-B provides 78.4%, a **+9.9**% accuracy boost over ViT-B under *identical settings*, while having 2.6×/2.4×fewer FLOPs/parameters. When changing the frame sampling from 16×4 to 32×3 performance increases to 80.2%. Finally, we take this model and fine-tune it for 5 epochs with longer 64 frame input, after interpolating the temporal positional embedding, to reach **81.2**% top-1 using 3 spatial and 3 temporal views for inference (it is sufficient test with fewer

| model | pretrain | top-1 | top-5 | FLOPs×views | Param |
|---|---|---|---|---|---|
| TSM-RGB [178] | IN-1K+K400 | 63.3 | 88.2 | 62.4×3×2 | 42.9 |
| MSNet [155] | IN-1K | 64.7 | 89.4 | 67×1×1 | 24.6 |
| TEA [171] | IN-1K | 65.1 | 89.9 | 70×3×10 | - |
| ViT-B-TimeSformer [22] | IN-**21K** | 62.5 | - | 1703×3×1 | 121.4 |
| ViT-B (our baseline) | IN-**21K** | 63.5 | 88.3 | 180×3×1 | 87.2 |
| SlowFast R50, 8×8 [77] | K400 | 61.9 | 87.0 | 65.7×3×1 | 34.1 |
| SlowFast R101, 8×8 [77] | | 63.1 | 87.6 | 106×3×1 | 53.3 |
| **MViT**-B, 16×4 | | 64.7 | 89.2 | 70.5×3×1 | 36.6 |
| **MViT**-B, 32×3 | | 67.1 | 90.8 | 170×3×1 | 36.6 |
| **MViT**-B, 64×3 | | **67.7** | **90.9** | 455×3×1 | 36.6 |
| **MViT**-B, 16×4 | K600 | 66.2 | 90.2 | 70.5×3×1 | 36.6 |
| **MViT**-B, 32×3 | | 67.8 | 91.3 | 170×3×1 | 36.6 |
| **MViT**-B-24, 32×3 | | **68.7** | **91.5** | 236×3×1 | 53.2 |

Table 4.6: **Comparison with previous work on SSv2**.

temporal views if a clip has more frames). Further quantitative and qualitative results are in §A.2.

**Kinetics-600** [36] is a larger version of Kinetics. Results are in Table 5.3. We train MViT from-scratch, without any pre-training. **MViT**-B, 16×4 achieves 82.1% top-1 accuracy. We further train a deeper 24-layer model with longer sampling, **MViT**-B-24, 32×3, to investigate model scale on this larger dataset. MViT achieves state-of-the-art of 83.4% with 5-clip center crop testing while having 56.0× fewer FLOPs and 8.4× fewer parameters than ViT-L-ViViT [8] which relies on large-scale ImageNet-21K pre-training.

**Something-Something-v2** (SSv2) [97] is a dataset with videos containing object interactions, which is known as a 'temporal modeling' task. Table 4.6 compares our method with the state-of-the-art. We first report a simple ViT-B (our baseline) that uses ImageNet-21K pre-training. Our **MViT**-B with 16 frames has 64.7% top-1 accuracy, which is better than the SlowFast R101 [77] which shares the same setting (K400 pre-training and 3×1 view testing). With more input frames, our **MViT**-B achieves 67.7% and the deeper **MViT**-B-24 achieves 68.7% using our K600 pre-trained model of above. In general, Table 4.6 verifies the capability of temporal modeling for MViT.

**Charades** [241] is a dataset with longer range activities. We validate our model in Table 4.7. With similar FLOPs and parameters, our **MViT**-B 16×4 achieves better results (+2.0 mAP) than SlowFast R50 [77]. As shown in the Table, the performance of **MViT**-B is further improved by increasing the number of input frames and **MViT**-B layers and using K600 pre-trained models.

**AVA** [103] is a dataset with for spatiotemporal-localization of human actions. We validate our MViT on this detection task. Details about the detection architecture of MViT can be found in §A.5. Table 4.8 shows the results of our MViT models compared with SlowFast [77] and X3D [75]. We observe that MViT-B can be competitive to SlowFast and X3D using the same pre-training and testing strategy.

| model | pretrain | mAP | FLOPs×views | Param |
|---|---|---|---|---|
| Nonlocal [277] | IN-1K+K400 | 37.5 | 544×3×10 | 54.3 |
| STRG +NL [276] | | 39.7 | 630×3×10 | 58.3 |
| Timeception [134] | K400 | 41.1 | N/A×N/A | N/A |
| LFB +NL [287] | | 42.5 | 529×3×10 | 122 |
| SlowFast 50, 8×8 [77] | | 38.0 | 65.7×3×10 | 34.0 |
| SlowFast 101+NL, 16×8 [77] | | 42.5 | 234×3×10 | 59.9 |
| X3D-XL [75] | | 43.4 | 48.4×3×10 | 11.0 |
| **MViT**-B, 16×4 | | 40.0 | 70.5×3×10 | 36.4 |
| **MViT**-B, 32×3 | | 44.3 | 170×3×10 | 36.4 |
| **MViT**-B, 64×3 | | **46.3** | 455×3×10 | 36.4 |
| SlowFast R101+NL, 16×8 [77] | K600 | 45.2 | 234×3×10 | 59.9 |
| X3D-XL [75] | | 47.1 | 48.4×3×10 | 11.0 |
| **MViT**-B, 16×4 | | 43.9 | 70.5×3×10 | 36.4 |
| **MViT**-B, 32×3 | | 47.1 | 170×3×10 | 36.4 |
| **MViT**-B-24, 32×3 | | **47.7** | 236×3×10 | 53.0 |

Table 4.7: **Comparison with previous work on Charades**.

| model | pretrain | val mAP | FLOPs | Param |
|---|---|---|---|---|
| SlowFast, 4×16, R50 [77] | K400 | 21.9 | 52.6 | 33.7 |
| SlowFast, 8×8, R101 [77] | | 23.8 | 138 | 53.0 |
| **MViT**-B, 16×4 | | 24.5 | 70.5 | 36.4 |
| **MViT**-B, 32×3 | | 26.8 | 170 | 36.4 |
| **MViT**-B, 64×3 | | **27.3** | 455 | 36.4 |
| SlowFast, 8×8 R101+NL [77] | K600 | 27.1 | 147 | 59.2 |
| SlowFast, 16×8 R101+NL [77] | | 27.5 | 296 | 59.2 |
| X3D-XL [75] | | 27.4 | 48.4 | 11.0 |
| **MViT**-B, 16×4 | | 26.1 | 70.5 | 36.3 |
| **MViT**-B, 32×3 | | 27.5 | 170 | 36.4 |
| **MViT**-B-24, 32×3 | | **28.7** | 236 | 52.9 |

Table 4.8: **Comparison with previvous work on AVA v2.2**. All methods use *single center crop* inference following [75].

## Ablations on Kinetics

We carry out ablations on Kinetics-400 (K400) using 5-clip center 224×224 crop testing. We show top-1 accuracy (Acc), as well as computational complexity measured in GFLOPs for a single clip input of spatial size $224^2$. Inference computational cost is proportional as a fixed number of 5 clips is used (to roughly cover the inferred videos with $T\times\tau$=16×4 sampling.) We also report Parameters in M($10^6$) and training GPU memory in G($10^9$) for a batch size of 4. By default all MViT ablations are with **MViT**-B, $T\times\tau$=16×4 and max-pooling in MHSA.

| model | shuffling | FLOPs (G) | Param (M) | Acc |
|---|---|---|---|---|
| **MViT**-B | | 70.5 | 36.5 | **77.2** |
| **MViT**-B | ✓ | | | 70.1 (−7.1) |
| ViT-B | | 179.6 | 87.2 | 68.5 |
| ViT-B | ✓ | | | 68.4 (−0.1) |

Table 4.9: **Shuffling frames in inference. MViT**-B severely drops ($-7.1\%$) for shuffled temporal input, but ViT-B models appear to *ignore* temporal information as accuracy remains similar ($-0.1\%$).

**Frame shuffling.** Table 4.9 shows results for randomly shuffling the input frames in time during testing. All models are trained without any shuffling and have temporal embeddings. We notice that our **MViT**-B architecture suffers a significant accuracy drop of **-7.1**% (77.2 $\rightarrow$ 70.1) for shuffling inference frames. By contrast, ViT-B is surprisingly robust for shuffling the temporal order of the input.

This indicates that a naïve application of ViT to video does not model temporal information, and the temporal positional embedding in ViT-B seems to be fully ignored. We also verified this with the 79.3% ImageNet-21K pre-trained ViT-B of Table 5.2, which has ***the same accuracy*** of 79.3% for shuffling test frames, suggesting that it implicitly performs bag-of-frames video classification in Kinetics.

| variant | [$N_1$, $N_2$] | FLOPs (G) | Mem (G) | Acc |
|---|---|---|---|---|
| ViT-B | [12, 0] | 179.6 | 16.8 | 68.5 |
| 2-scale ViT-B, $Q$ pool | [6, 6] | **111.1** (−68.5) | 9.8 (−7.0) | **71.0** (+1.5) |
| ViT-B, $K, V$ pool | [12, 0] | 148.4 (−31.2) | **8.9** (−7.9) | 69.1 (+0.6) |

Table 4.10: **Query (scale stage) and Key-Value pooling on ViT-B.** Introducing a *single* extra resolution stage into ViT-B boosts accuracy by +1.5%. Pooling $K, V$ provides +0.6% accuracy. Both techniques allow dramatic FLOPs/memory savings.

**Two scales in ViT.** We provide a simple experiment that ablates the effectiveness of scale-stage design on ViT-B. For this we add a *single scale stage* to the ViT-B model. To isolate the effect of having different scales in ViT, we do not alter the channel dimensionality for this experiment. We do so by performing $Q$-Pooling with $s^Q \equiv (1, 2, 2)$ after 6 Transformer blocks (*cf.* Table A.1). Table 4.10 shows the results. Adding a single scale stage to the ViT-B baseline boosts accuracy by +1.5% while deceasing FLOPs and memory cost by 38% and 41%. Pooling Key-Value tensors reduces compute and memory cost while slightly increasing accuracy.

**Separate space & time embeddings in MViT.** In Table 4.11, we ablate using (i) none, (ii) space-only, (iii) joint space-time, and (iv) a separate space and time (our default), positional embeddings. We observe that no embedding (i) decays accuracy by -0.9% over using just a spatial one (ii) which

| | positional embedding | Param (M) | Acc |
|---|---|---|---|
| (i) | none | 36.2 | 75.8 |
| (ii) | space-only | 36.5 | 76.7 |
| (iii) | joint space-time | 38.6 | 76.5 |
| (iv) | **separate** in space & time | 36.5 | **77.2** |

Table 4.11: **Effect of separate space-time positional embedding.** Backbone: **MViT**-B, 16×4. FLOPs are 70.5G for all variants.

is roughly equivalent to a joint spatiotemporal one (iii). Our separate space-time embedding (iv) is best, and also has 2.1M fewer parameters than a joint spacetime embedding.

| $T \times \tau$ | $c_T \times c_H \times c_W$ | $s_T \times s_H \times s_W$ | FLOPs | Param | Acc |
|---|---|---|---|---|---|
| 8×8 | 1×4×4 | 1×4×4 | 69.4 | 36.5 | 74.5 |
| 8×8 | 1×7×7 | 1×4×4 | 69.6 | 36.5 | 75.6 |
| 8×8 | 3×7×7 | 1×4×4 | 70.5 | 36.5 | 75.9 |
| <u>16×4</u> | <u>3×7×7</u> | <u>2×4×4</u> | 70.5 | 36.5 | 77.2 |
| 32×2 | 3×7×7 | 4×4×4 | 70.5 | 36.5 | 77.2 |
| 32×2 | 7×7×7 | 4×4×4 | 70.5 | 36.5 | 77.3 |

Table 4.12: **Input sampling**: We vary sampling rate $T \times \tau$, the size $\mathbf{c}$=$c_T \times c_H \times c_W$ and stride of $\mathbf{s}$=$s_T \times s_H \times s_W$ the cube$_1$ layer that projects space-time cubes. Cubes with temporal extent $c_T > 1$ are beneficial. Our default setting is <u>underlined</u>.

**Input Sampling Rate.** Table 4.12 shows results for different cubification kernel size $\mathbf{c}$ and sampling stride $\mathbf{s}$ (*cf.* Table 4.2). We observe that sampling *patches*, $\mathbf{c}_T$ = 1, performs worse than sampling *cubes* with $\mathbf{c}_T > 1$. Further, sampling twice as many frames, $T$= 16, with twice the cube stride, $s_T$= 2, keeps the cost constant but boosts performance by +1.3% (75.9% $\rightarrow$ 77.2%). Also, sampling *overlapping* input cubes $\mathbf{s} < \mathbf{c}$ allows better information flow and benefits performance. While $\mathbf{c}_T > 1$ helps, very large temporal kernel size ($\mathbf{c}_T = 7$) does not further improve performance.

| variant | [$N_2$, $N_3$, $N_4$, $N_5$] | FLOPs | Param | Mem | Acc |
|---|---|---|---|---|---|
| V1 | [2, 6, 6, 2] | 90.2 | **29.5** | 11.0 | 76.3 |
| V2 | [2, 4, 6, 4] | 86.9 | 42.8 | 10.3 | 75.9 |
| V3 | [2, 4, 8, 2] | 88.3 | 32.2 | 10.5 | 76.6 |
| V4 | [2, 2, 8, 4] | 85.0 | 45.5 | 9.7 | 76.7 |
| <u>V5</u> | <u>[1, 2, 11, 2]</u> | **83.6** | 36.5 | **9.1** | **77.1** |
| V6 | [2, 2, 10, 2] | 86.4 | 34.9 | 11.3 | 76.9 |

Table 4.13: **Scale blocks**: We ablate the stage configuration as the number of blocks $N$ in stages of **MViT**-B (*i.e.* where to pool $Q$). The overall number of transformer blocks is constant with $N$=16.

**Stage distribution.** The ablation in Table 4.13 shows the results for distributing the number of transformer blocks in each individual scale stage. The overall number of transformer blocks, $N{=}16$ is consistent. We observe that having more blocks in early stages increases memory and having more blocks later stages the parameters of the architecture. Shifting the majority of blocks to the scale$_4$ stage (Variant V5 and V6 in Table 4.13) achieves the best trade-off.

| stride s | adaptive | FLOPs | Mem | Acc |
|---|---|---|---|---|
| none | n/a | 130.8 | 16.3 | **77.6** |
| 1×4×4 | | 71.4 | 8.2 | 75.9 |
| 2×4×4 | | 64.3 | 6.6 | 74.8 |
| 2×4×4 | ✓ | 83.6 | 9.1 | 77.1 |
| 1×8×8 | ✓ | 70.5 | 6.8 | 77.2 |
| 2×8×8 | ✓ | **63.7** | **6.3** | 75.8 |

Table 4.14: **Key-Value pooling**: Vary stride $\mathbf{s} = s_T \times s_H \times s_W$, for pooling $K$ and $V$. "adaptive" reduces stride w.r.t. stage resolution.

**Key-Value pooling.** The ablation in Table 4.14 analyzes the pooling stride $\mathbf{s} = s_T \times s_H \times s_W$, for pooling $K$ and $V$ tensors. Here, we compare an "adaptive" pooling that uses a stride w.r.t. stage resolution, and keeps the $K, V$ resolution *fixed* across all stages, against a non-adaptive version that uses the same stride at every block. First, we compare the baseline which uses no $K, V$ pooling with non-adaptive pooling with a fixed stride of 2×4×4 across all stages: this drops accuracy from 77.6% to 74.8 (and reduces FLOPs and memory by over 50%). Using an adaptive stride that is 1×8×8 in the scale$_1$ stage, 1×4×4 in scale$_2$, and 1×2×2 in scale$_3$ gives the best accuracy of 77.2% while still preserving most of the efficiency gains in FLOPs and memory.

| kernel $\mathbf{k}$ | pooling func | Param | Acc |
|---|---|---|---|
| s | max | 36.5 | 76.1 |
| $2\mathbf{s}+1$ | max | 36.5 | 75.5 |
| $\mathbf{s}+1$ | max | 36.5 | 77.2 |
| $\mathbf{s}+1$ | average | 36.5 | 75.4 |
| $\mathbf{s}+1$ | conv | 36.6 | 78.3 |
| 3×3×3 | conv | 36.6 | **78.4** |

Table 4.15: **Pooling function**: Varying the kernel $\mathbf{k}$ as a function of stride $\mathbf{s}$. Functions are average or max pooling and conv which is a learnable, channel-wise convolution.

**Pooling function.** The ablation in Table 4.15 looks at the kernel size $\mathbf{k}$ w.r.t. the stride $\mathbf{s}$, and the pooling function (max/average/conv). First, we see that having equivalent kernel and stride $\mathbf{k}{=}\mathbf{s}$ provides 76.1%, increasing the kernel size to $\mathbf{k}{=}2\mathbf{s}+1$ decays to 75.5%, but using a kernel $\mathbf{k}{=}\mathbf{s}+1$ gives a clear benefit of 77.2%. This indicates that *overlapping pooling is effective*, but a too large overlap $(2\mathbf{s}+1)$ hurts. Second, we investigate average instead of max-pooling and observe that accuracy decays by from 77.2% to 75.4%.

Third, we use conv-pooling by a learnable, channelwise convolution followed by LN. This variant has +1.2% over max pooling and is used for all experiments in §4.4 and §4.5.

| model | clips/sec | Acc | FLOPs×views | Param |
|---|---|---|---|---|
| X3D-M [75] | 7.9 | 74.1 | 4.7×1×5 | 3.8 |
| SlowFast R50 [77] | 5.2 | 75.7 | 65.7×1×5 | 34.6 |
| SlowFast R101 [77] | 3.2 | 77.6 | 125.9×1×5 | 62.8 |
| ViT-B [67] | 3.6 | 68.5 | 179.6×1×5 | 87.2 |
| **MViT**-S, max-pool | **12.3** | 74.3 | 32.9×1×5 | 26.1 |
| **MViT**-B, max-pool | 6.3 | 77.2 | 70.5×1×5 | 36.5 |
| **MViT**-S, conv-pool | 9.4 | 76.0 | 32.9×1×5 | 26.1 |
| **MViT**-B, conv-pool | 4.8 | 78.4 | 70.5×1×5 | 36.6 |

Table 4.16: **Speed-Accuracy tradeoff on Kinetics-400.** Training throughput is measured in clips/s. MViT is *fast* and *accurate*.

**Speed-Accuracy tradeoff.** In Table 4.16, we analyze the speed/accuracy trade-off of our MViT models, along with their counterparts vision transformer (ViT [67]) and ConvNets (SlowFast 8×8 R50, SlowFast 8×8 R101 [77], & X3D-L [75]). We measure training throughput as the number of video clips per second on a single M40 GPU.

We observe that both MViT-S and MViT-B models are not only significantly more accurate but also much faster than both the ViT-B baseline and convolutional models. Concretely, MViT-S has **3.4**× higher throughput speed (clips/s), is **+5.8**% more accurate (Acc), and has **3.3**× fewer parameters (Param) than ViT-B. Using a conv instead of max-pooling in MHSA, we observe a training speed reduction of ~20% for convolution and additional parameter updates.

## 4.5 Experiments: Image Recognition

We apply our video models on static image recognition by using them with single frame, $T = 1$, on ImageNet-1K [57].

**Training.** Our recipe is identical to DeiT [264] and summarized in §A.5. Training is for 300 epochs and results improve for training longer [264].

### Main Results

For this experiment, we take our models which were designed by ablation studies for video classification on Kinetics and *simply remove the temporal dimension*. Then we train and validate them (“from scratch”) on ImageNet.

Table 5.1 shows the comparison with previous work. From top to bottom, the table contains RegNet [221] and EfficientNet [255] as ConvNet examples, and DeiT [264], with DeiT-B being identical to ViT-B [67] but trained with the improved recipe in [264]. Therefore, this is the vision transformer counterpart we are interested in comparing to.

The bottom section in Table 5.1 shows results for our Multiscale Vision Transformer (MViT) models.

| model | Acc | FLOPs (G) | Param (M) |
|---|---|---|---|
| RegNetZ-4GF [65] | 83.1 | 4.0 | 28.1 |
| RegNetZ-16GF [65] | 84.1 | 15.9 | 95.3 |
| EfficientNet-B7 [255] | 84.3 | 37.0 | 66.0 |
| DeiT-S [264] | 79.8 | 4.6 | 22.1 |
| DeiT-B [264] | 81.8 | 17.6 | 86.6 |
| DeiT-B $\uparrow 384^2$ [264] | 83.1 | 55.5 | 87.0 |
| **MViT**-B-16, max-pool | 82.5 | 7.8 | 37.0 |
| **MViT**-B-24, max-pool | 83.1 | 11.0 | 53.5 |
| **MViT**-B-24-wide-$320^2$, max-pool | 84.3 | 32.7 | 72.9 |
| **MViT**-B-16 | 83.0 | 7.8 | 37.0 |
| **MViT**-B-24-wide-$320^2$ | **84.8** | 32.7 | 72.9 |

Table 4.17: **Comparison to prior work on ImageNet**. RegNet and EfficientNet are ConvNet examples that use different training recipes. DeiT/MViT are ViT-based and use identical recipes [264].

We show models of different depth, **MViT**-B-Depth, (16, 24, and 32), where **MViT**-B-16 is our base model and the deeper variants are simply created by repeating the number of blocks $N_*$ in each scale stage (*cf.* Table A.1b). "wide" denotes a larger channel dimension of $D = 112$. All our models are trained using the identical recipe as DeiT [264].

We make the following observations:

(i) Our lightweight **MViT**-B-16 achieves 82.5% top-1 accuracy, with only 7.8 GFLOPs, which outperforms the DeiT-B counterpart by +0.7% with lower computation cost (2.3×fewer FLOPs and Parameters). If we use conv instead of max-pooling, this number is increased by +0.5% to 83.0%.

(ii) Our deeper model **MViT**-B-24, provides a gain of +0.6% accuracy at slight increase in computation.

(iii) A larger model, **MViT**-B-24-wide with input resolution $320^2$ reaches 84.3%, corresponding to a +1.2% gain, at 1.7×fewer FLOPs, over DeiT-B$\uparrow 384^2$. Using convolutional, instead of max-pooling elevates this to **84.8**%.

These results suggest that Multiscale Vision Transformers have an architectural advantage over Vision Transformers.

## 4.6 Conclusion

We have presented Multiscale Vision Transformers that aim to connect the fundamental concept of multiscale feature hierarchies with the transformer model. MViT hierarchically expands the feature complexity while reducing visual resolution. In empirical evaluation, MViT shows a fundamental advantage over single-scale vision transformers for video and image recognition. We hope that our approach will foster further research in visual recognition.

# Chapter 5

# Reversible Vision Transformers

## 5.1 Introduction

The deep learning revolution in computer vision has rested on the bedrock of high performance hardware accelerators. Fueled by special purpose AI accelerators, the compute requirements for state-of-the-art models are growing exponentially. However, compute is only half the story. The other, and often overlooked half, is memory bandwidth bottleneck, which has been difficult to proportionally scale as compared to peak accelerator FLOPs [212]. In particular, the peak accelerator FLOPs have been increasing at a rate of ~3.1× every 2 years [87, 251]. However, peak bandwidth only scales at a rate of ~1.4× every 2 years. This disparity is exacerbated in transformers, which have been doubling in required compute roughly every three months for the past three years, resulting in a so-called memory wall [87] where both the overall model performance as well as the training speed have become tightly memory-bound [135].

As such, for bandwidth bound models, trading compute for memory through re-computation could actually be more efficient than using work-optimal algorithms [283, 282]. In the case of training neural network models, this can be achieved by re-computing activations instead of storing and then loading them from DRAM [125]. Besides training speed, scaling vision transformers in depth naturally hits the GPU memory capacity, especially in memory starved regimes such as video recognition where state-of-the-art models are often limited to batch size 1 due to high memory footprint of intermediate activations.

We propose Reversible Vision Transformers, a family of expressive visual recognition architectures with very favorable activation memory footprints (Figure 5.1) compared to their non-reversible variants. By trading-off GPU activation caching with efficient on-the-fly activation re-computation, reversible vision transformers effectively *decouple* the activation *memory* growth from the *depth* of the model.

This chapter is based on joint work with Haoqi Fan, Chao-Yuan Wu, Bo Xiong, Christoph Feichtenhofer and Jitendra Malik [197], and is presented as it appeared in the CVPR 2022 proceedings.

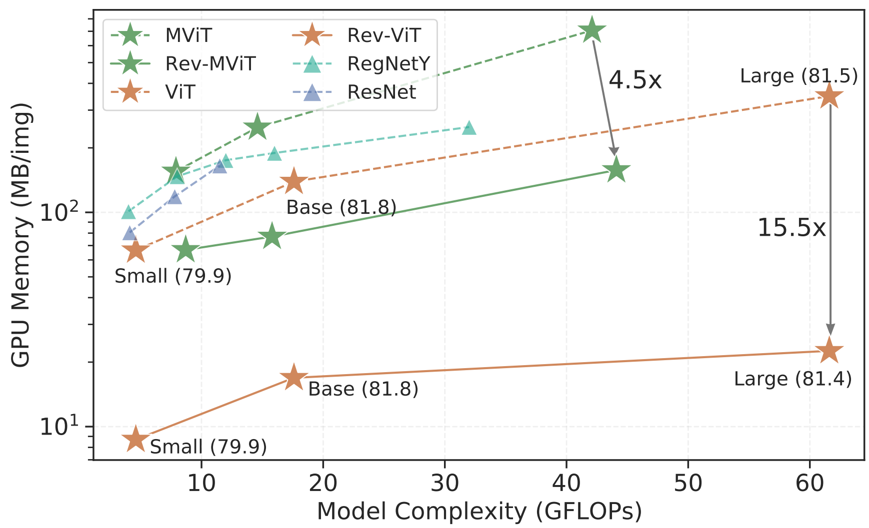



Figure 5.1: **Reversible Vision Transformers** are more memory-efficient, yet powerful *reversible counterparts* of state-of-the-art Vision Transformer (ViT) [68] and Multiscale Vision Transformer (MViT) [71] architectures with varying model complexity. Numbers in parentheses denote top-1 ImageNet performance. ResNet [111] and RegNet [221] are only shown for reference. For detailed discussion please refer to Sec. 5.4.

While the natural language processing community has performed some early exploration of reversible transformers for machine translation [149], these techniques focus on *longer sequence lengths* rather than depth.

Our experiments show that a straightforward adaptation of vision transformers to reversible architectures *fails* to scale for *deeper* models because of training convergence instabilities due to internal sub-block residual connections.

In this work, we reconfigure the residual paths in Vision Transformers (ViT) [68] and Multiscale Vision Transformers (MViT) [71] to overcome this issue. We further find that reversible structures have stronger inherent regularization and therefore, we use a lighter augmentation recipe (repeated augmentation, augmentation magnitude and stochastic depth) and lateral connections between residual blocks.

We benchmark extensively across image recognition tasks such as image classification and object detection as well as video classification, across all of which, reversible vision transformers have competitive performance to their non-reversible counterparts suffering negligible to no performance decay. Moreover, reversible models have extremely favorable per-image memory footprint, saving **15.5**$\times$ on the ViT-Large model and **4.5**$\times$ on the MViT-Large model with reversible training.

In summary, our contributions are three-fold.

(*i*) We propose Reversible Vision Transformer (**Rev-ViT**) and Reversible Multiscale Vision Transformers (**Rev-MViT**), memory efficient reversible adaptations of state-of-the-art visual recognition backbones.

(*ii*) We observe reversible transformers to have a stronger inherent regularization than vanilla networks. Hence, we develop new training recipes by adapting the original recipes with different repeated augmentations, augmentation magnitudes and drop path rate to match the performance of their non-reversible counterparts.

(*iii*) We benchmark our models across several tasks: image classification, object detection and action recognition, across accuracy, memory, maximum training batch size and model complexity. In particular, at matched complexity (FLOPs/parameters) and final accuracy, Rev-ViT-B and Rev-ViT-L train with per image memory footprints that are 8.2× and 15.5× lighter than ViT-B and ViT-L respectively. Further, we show how *deep* reversible networks can achieve up to 2-4× throughput than their vanilla counterparts.

## 5.2 Related Work

**Transformers** are a popular network structure that were first proposed for natural language applications [270] and now are widely used in all areas of deep learning such as Reinforcement Learning [44], Speech [167], Music [129], multi-modal learning [137] and recently, in traditional vision tasks [68] as well. Since their introduction, Vision Transformers have experienced enthusiastic adoption and have been applied to several visual recognition tasks [68, 262, 263] using priors such as multi-scale feature hierarchies [71, 99, 187, 275, 310] and local structure modelling [187, 66, 47]. Further, vision transformers have also been generalized for action recognition and detection in videos [71, 187, 9, 211, 204, 22].

However, a crucial problem with scaling up transformer models is the growth of required GPU memory with depth. This linear growth in memory is prohibitive to the development of very deep models since the batch size needs to be reduced considerably to be able to accommodate storing the intermediate activations on GPU. This problem is exacerbated in video models which process very large input tensors and are often trained with batch size 1 even for shallower depths. A potential systems-level solution to scale up conventional transformer architectures is model parallelism [56] that puts different parts of the model on different GPUs. However in practice, it is quite slow and requires special high bandwith network infrastructure because of huge across device traffic.

In this work, we use Vision Transformers [68] and Multiscale Vision Transformers [71] as our base models and propose their reversible transformer version that decouple the memory requirement from depth of the model. This facilitates saving GPU memory and allows training with much higher batch size, and consequently, to preserve or even *increase* training throughput of deep non-reversible models.

**Reversible Architectures** are a family of neural network architectures that are based on the NICE [63, 64] reversible transformation model which are the precursors of the modern day generative flow based image generation architectures [122, 147]. Based on the NICE invertible transformations, Gomez *et al*. [91] propose a Reversible ResNet architecture that employs the reversible transformation [63] for memory-efficient image classification in ResNets [112]. An interesting line of work builds upon the Reversible ResNets ideas proposing better reversible CNN models using ODE characterizations [41, 162, 237], momentum [162, 237], layer-wise inversion [110], fourier transform based inversion [79] and fixed point iteration based inversion [17, 247]. Reversible CNNs have been applied to several traditional image tasks such as compression [183], reconstruction [170], retrieval [169], and denoising [131, 185] as well as to compressed sensing [249], compact resolution [302], image to image translation [208], remote sensing [216], medical image segmentation [215, 299] and MRI reconstruction [218]. Reversible transformation have also been adapted to other networks such as RNNs [192], Unet [28, 70], Masked Convolutional Networks [247] and 1000-layer deep Graph Neural Networks [164]. Some early attempts have also been made to adapt the reversible transformation to the NLP domain, initiated by Kiatev *et al*. [149] and built upon in [323, 324] for machine translation.

However, word-level input partitioning contains much richer semantic content than patch level image partitioning and NLP transformers tend to be shallower in depth but wider in channel dimension. For example, Kiatev *et al*. [149] focus on expanding on the input sequence dimension rather than model depth and with no benchmarking on maximum batch-size, peak GPU memory and training throughput.

Our experiments show that a naïve adaption of reversible vision transformers performs poorly for deeper ($\geq$8 blocks) models. This work is the first to propose Reversible Vision Transformers, adapt it to two state-of-the-art transformer networks, namely, ViT and MViT. Furthermore, this work is the first use of a reversible backbone for object detection and video classification, which tends to be one the most memory starved domains of visual recognition.

## 5.3 Approach

We first present a brief overview of the reversible transformation (Sec. 5.3) and its benefits in neural network training (Sec. 5.3). We then present our proposed Reversible Vision Transformer (Sec. 5.3) its two residual stream structure (Sec. 5.3 and associated constraints (Sec. 5.3. This is followed by our proposed Reversible Multiscale Vision Transformer architecture (Sec. 5.3) and its sub-blocks (Sec. 5.3 and Sec. 5.3) that allow end-to-end reversible training.

### Reversible Block Structure

The reversible transformer is composed of a stack of reversible blocks that follow the structure of the reversible transformation to allow analytic invertibility of outputs.

#### Reversible Transformation

Consider a transformation $T_1$ that transforms an input tensor $I$ partitioned into two $d$ dimensional tensors, $[I_1; I_2]$ into the output tensor $O$ also similarly partitioned into tensors, $[O_1; O_2]$ with an arbitrary differentiable function $F(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ as follows:

$$\mathbf{I} = \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} \xrightarrow[T_1]{} \begin{bmatrix} O_1 \\ O_2 \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 + F(I_1) \end{bmatrix} = \mathbf{O}$$

Note that the above transformation $T_1$ allows an inverse transformation $T_1'$ such that $T_1' \circ T_1$ is an identity transform. Also, consider an analogous transposed transformation $T_2$ using the function $G(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ as follows:

$$\mathbf{I} = \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} \xrightarrow[T_2]{} \begin{bmatrix} O_1 \\ O_2 \end{bmatrix} = \begin{bmatrix} I_1 + G(I_2) \\ I_2 \end{bmatrix} = \mathbf{O}$$

Similar to $T_1$, $T_2$ also allows an inverse transform $T_2'$. Now consider the composition $T = T_2 \circ T_1$ that transforms both the partitions of the input vector $\mathbf{I}$ and is obtained as,

$$\mathbf{I} = \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} \xrightarrow[T]{} \begin{bmatrix} O_1 \\ O_2 \end{bmatrix} = \begin{bmatrix} I_1 + G(I_2 + F(I_1)) \\ I_2 + F(I_1) \end{bmatrix} = \mathbf{O} \tag{5.1}$$

Naturally, $T$ affords the inverse transform $T' = T_1' \circ T_2'$ that follows $T'(T(\mathbf{I})) = \mathbf{I}$. Note that the inverse transform $T'$ queries the functions $F$ and $G$ exactly once and hence has the same computational cost as the forward transform $T$.

### Vanilla networks require caching activations

Consider the back-propagation mechanism. Given a computation graph node, $\mathcal{M}$, its children nodes $\{\mathcal{N}_j\}$, and the gradients of the children node with respect to final loss $\left\{\frac{d\mathcal{L}}{d\mathcal{N}_j}\right\}$, the back-propagation algorithm uses the chain rule to calculate the gradient with respect to $\mathcal{M}$ as,

$$\frac{d\mathcal{L}}{d\mathcal{M}} = \sum_{\mathcal{N}_j} \left(\frac{\partial f_j}{\partial \mathcal{M}}\right)^T \frac{d\mathcal{L}}{d\mathcal{N}_j}$$

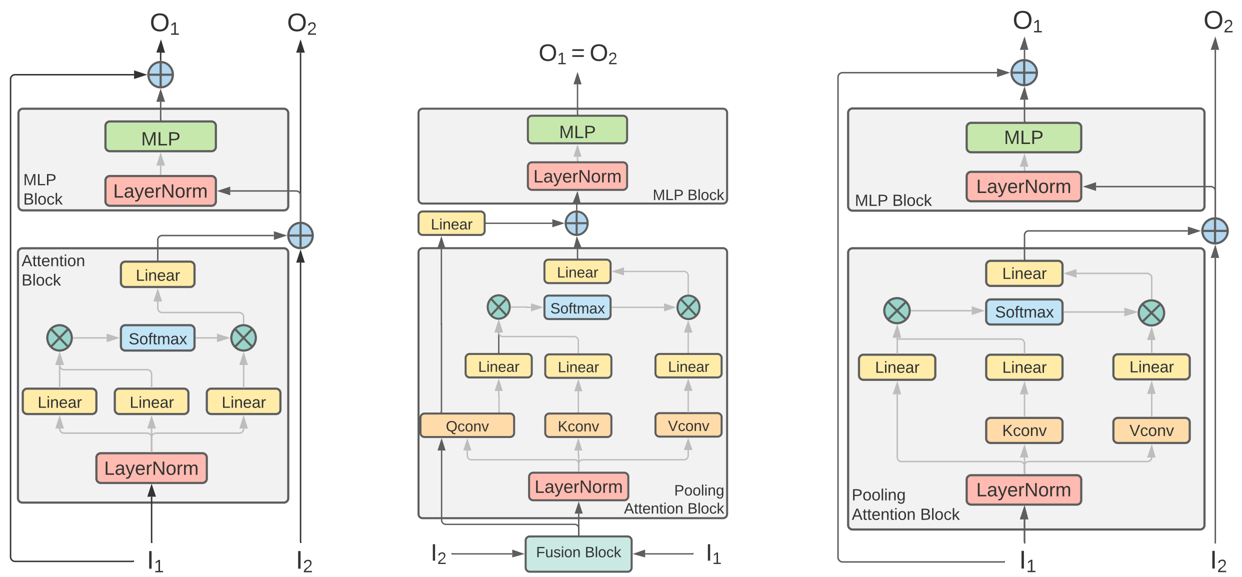
where $f_j$ denotes the function computing node $\mathcal{N}_j$ from its parents, $\mathcal{M}$ being one of them. The jacobian $\frac{\partial f_j}{\partial \mathcal{M}}$, requires calculating the partial gradient of the $f_j$ output with respect to the current node $\mathcal{M}$.

Now consider the simplest possible neural network layer $f(X) = W^T X$, where $X$ is an intermediate activation inside the network. Applying the above described backpropagation algorithm to compute the derivative with respect to parent nodes, and using the output $Y$ as the sole child node, $\mathcal{N}_j$, we get,

$$\frac{d\mathcal{L}}{dW} = \left(\frac{d\mathcal{L}}{dY}\right) X^T \qquad\qquad \frac{d\mathcal{L}}{dX} = W \frac{d\mathcal{L}}{dY}$$

Thus, because of the function jacobian, the backpropagation algorithm requires intermediate activations during the forward pass to be available in the backward pass to compute the gradients with respect to the weights.

Typically, this is achieved by caching the intermediate activations on GPU memory for use in the backward pass. This allows fast gradient computation at the cost of extra memory. Further, the sequential nature of the network requires the activations for all the layers to be cached in before the loss gradients are calculated and the cached memory is freed. This dependence significantly affects the peak memory usage which thus becomes linearly dependent on the network depth $D$.




(a) Rev-**ViT** Block (b) Stage-Transition Rev-**MViT** Block (c) Stage-Preserving Rev-**MViT** Block

Figure 5.2: **Reversible ViT** is a two-residual-stream architecture composed of a stack of Reversible ViT blocks (a) that transforms the inputs $I_1$ and $I_2$ with the ViT design [68], but in our reversible fashion. **Reversible MViT** is a two-residual-stream architecture as well, made up of a stack of two type of blocks – (b) The stage-transition blocks that act as coupling between the residual streams as well as perform channel upsampling and resolution downsampling and (c) the stage-preserving blocks that form the majority of the computational graph and propagate information preserving input feature dimension.

### Learning without caching activations

As noted in Sec. 5.3, an input transformed with the reversible transformation $T$ allows recalculating the input from the output of the transformation. Hence, a network composed of such reversible transformations *does not need to store intermediate activations* since they can be recomputed easily in the backward pass from the output. However the reversible transformation $T$ places an important constraint on the property of the learnt function.

**Equidimensional Constraint**. As mentioned in Sec. 5.3, the functions $F$ and $G$ need to be equidimensional in input and output spaces. Hence, the feature dimensions need to remain constant under $T$. While this constraint is an obstruction for other vision architectures such as ResNets [112] that require a change of feature dimensions, it is easily satisfied in the Vision Transformer architecture [68] which maintains a constant feature dimension throughout the layers.

## Reversible Vision Transformers

### Adapting ViT to Two-Residual-Streams

Fig. 5.2a shows the reversible transformation $T$ adapted to the Vision Transformer architecture [68]. The input consists of two partitioned tensors $I_1$ and $I_2$ that are transformed as per the equation 5.3 maintaining reversibility. This leads to a *two-residual-stream* architecture where each of the inputs $I_1$ and $I_2$ maintain their own residual streams while mixing information with each other using functions $F$ and $G$. Following ViT [68], we use the Multi-head attention and the MLP sublocks as functions $F$ and $G$ respectively.

### Boundary Conditions

As the ViT architecture only uses a single residual stream, the architecture needs to be modified to support the two-residual-stream design (Sec. 5.3). We propose the following:
**1. Initiation.** We keep the stem intact and send the patchification output activations to $I_1$ and $I_2$. Note that this design choice is different from [92] which proposes to split in halves along the channel dimensions.
**2. Termination.** The two residual paths need to be fused before the final classifier head to preserve information. We propose to layer-normalize the inputs first, followed by concatenation, to reduce the fusion computational overhead.

### Reconfiguring Residual Connections

Residual connections play a key role for signal propagation in deep networks [112]. The reversible transform $T$ itself also depends crucially on the residual connections between the two streams to maintain reversibility. Interestingly, we observe a key relationship between the residual connections and signal propagation in Reversible Vision Transformer.

Note that while it is common practice for neural network blocks to be wrapped around a residual connection for better gradient flow [112], there is no such connection for either the $I_1$ or $I_2$ inputs. Specifically, internal residual connections around the MLP and attention sub-blocks for both the $I_1$ and $I_2$ streams are absent. Instead, the residual connections for each residual stream flows through the other stream, operating through the inherent skip connection present in the reversible transformation $T$ (Sec. 5.3). We find these *internal* skip connections *detrimental* to training convergence for deeper models while bringing no additional gain for shallower models and choose to omit them entirely for reversible vision transformer blocks.

## Reversible Multiscale Vision Transformers

The recently proposed MViT architecture develops a feature hierarchy inside the model by *down-sampling* the visual *resolution* and *upsampling* the *channel* dimension. It obtains state-of-the-art results on both image and video classification benchmarks. To showcase the flexibility of the reversible design, we adapt the MViT model to Reversible Multiscale Vision Transformers. We propose to compose the Reversible MViT architecture in the same structure as the MViT model but using two different layers – the *Stage Transition* and the *Stage-Preserving* blocks.

### Stage-Transition Block

Figure 5.2b depicts the architecture of the proposed stage-transition block. The stage-transition block closely follows the design of the resolution upsampling blocks in MViT [71] with the following crucial modifications:

**Lateral Connections.** The residual streams $I_1$ and $I_2$ are fused with lateral connections at the start of the stage-transition block. This allows efficient computation of the resolution downsampling and feature upsampling without repeat computation in each stream separately.

**Feature Upsampling.** MViT performs feature upsampling in the last MLP block before the resolution upsampling block. We propose to move the channel upsampling stage inside the pooling attention sub-block of the stage-transition block. Specifically, we propose to upsample the Query, Key and Value vectors in the linear layer following the pooling channel-wise convolutional layers (Figure 5.2b and 5.2c). This was the dual benefit of (A) allowing all feature dimension changes to take place in sync inside the same block and allowing other blocks to keep feature dimensions intact, a virtue of reversible architectures (Sec. 5.3) and (B) saving additional computation from being used in the prior MLP and pooling layers. We follow the same boundary conditions at the stage-transition blocks as in the reversible vision transformer architecture (Sec. 5.3).

### Stage-Preserving Block

Figure 5.2c shows the reversible transformation $T$ (Sec. 5.3) adapted to the Multiscale Vision Transformer architecture [71]. The design closely resembles that of the reversible vision transformer block (Figure 5.2a) with the addition of multi-head pooling attention [71]. Note that even though the attention uses pooling on key and value tensors, thereby changing the sequence length, the output dimensions are still preserved. Hence, the stage-preserving block still follows the equidimensional constraint (Sec. 5.3) and hence can be made fully reversible and learnt without caching activations.

Since each stage-transition block changes the spatiotemporal resolution, they occur only a limited number of times in the entire MViT network. In other words, the majority of the computation as well as memory usage is performed within the stage-preserving blocks and is fully reversible. We follow the same residual connection circuit (Sec. 5.3) as in Reversible Vision Transformer blocks for both the stage-transition and the stage-preserving blocks.

## 5.4 Results

**Datasets.** We benchmark both the Reversible Vision Transformer and the Reversible Multiscale Vision Transformer architectures extensively across image classification (ImageNet [58]), video classification (Kinetics 400 [143] & Kinetics 600 [34]) and object detection (MS-COCO [181]).

Across all the benchmarks, we observe significant memory savings by using the reversible architecture with negligible to no accuracy change. All presented results and ablations are trained from random initialization, except for COCO where we initialize from ImageNet weights.

## Image Classification

**Settings.** We benchmark our proposed models on image classification on the ImageNet-1K dataset [58] with ~1.28M images among 1000 classes. We follow training recipes [72] for both ViT [68] and MViT [71] models with certain crucial adaptions (Sec. 5.6). All models are trained from random initialization without EMA for 300 epochs except for ViT-L and Rev-ViT-L which follow a 200 epoch training recipe. Training details are in Supplementary.

**Results.** Table 5.1 shows the results for Reversible Vision and Reversible Multiscale Vision Transformers across different models and FLOP regimes. We benchmark all the models on a single 16 GB V100 GPU under $224 \times 224$ image size and otherwise identical conditions. The maximum batch size is obtained as the highest number of images in a batch than can train without running out of GPU memory. The memory per image is measured as the peak GPU memory each image occupies during training.

We note that Reversible Vision Transformers match the FLOP and parameter specifications of their non-reversible counterparts exactly owing to the parsimonious design of the reversible vision transformer block (Sec. 5.3). The Reversible Multiscale Vision Transformer has slightly higher FLOPs due to the stage-transition (Sec. 5.3) stages while still being very GPU memory efficient owing to the stage-preserving (Sec. 5.3) stages.

**Increasing memory savings with depth.** In Table 5.1, we observe that our Rev-ViT matches the performance of vanilla ViT to very close fidelity across all model variants (Small, Base and Large) and FLOP regimes. Since the memory used per image is linearly dependent on the depth of the model for vanilla networks (Sec. 5.3), the memory gains of the reversible model increases as the network scales in depth. Notably, while the Reversible ViT-S already enjoys an impressive memory saving of about **86.8**% (equivalent to a **7.6**$\times$ reduction) with respect to the vanilla ViT-S model, the gain increases further to **15.5**$\times$ or, about **93**% memory savings for the Reversible ViT-L model.

Equivalently, the saved memory can be used to increase the training batch size where we observe a similar trend as well. While reversible ViT-S model achieves a **6.1**$\times$ increase in batch size on the ViT-S model, the effect is more for ViT-L model where the maximum batch size increases by **14.3**$\times$ jumping from a small $24$ image per batch to $344$ images. This is a very favorable trend, since it is indeed the deeper models that hit the *GPU memory wall* [87].

Further, hierarchical vision transformers such as MViT also enjoy a memory saving of about **52.1**% without suffering any significant drop in performance. The memory savings in Rev-MViT are smaller compared to the ViT variants because of the stage-transition blocks in hierarchical models (Sec. 5.3) that require storing the input activations due to the non-reversible nature of pooling attention stemming from the feature dimension change [71].

| model | Acc | **Memory** (MB/img) | **Maxiumum Batch Size** | GFLOPs | Param (M) |
|---|---|---|---|---|---|
| ResNet-101 [113] | 76.4 | 118.7 | 112 | 7.6 | 45 |
| ResNet-152 [113] | 77.0 | 165.2 | 79 | 11.3 | 60 |
| RegNetY-4GF [221] | 80.0 | 101.1 | 136 | 4.0 | 21 |
| RegNetY-12GF [221] | 80.3 | 175.2 | 75 | 12.1 | 51.8 |
| RegNetY-32GF [221] | 80.9 | 250.2 | 46 | 32.3 | 32.3 |
| Swin-T [186] | 81.3 | - | - | 4.5 | 29 |
| ViT-S [262] | 79.9 | 66.5 | 207 | 4.6 | 22 |
| Rev-ViT-S | 79.9 | **8.8 $\downarrow$7.5$\times$** | **1232 $\uparrow$5.9$\times$** | 4.6 | 22 |
| ViT-B [262] | 81.8 | 129.7 | 95 | 17.6 | 87 |
| Rev-ViT-B | 81.8 | **17.0 $\downarrow$7.6$\times$** | **602 $\uparrow$6.3$\times$** | 17.6 | 87 |
| RegNetY-8GF [221] | 81.7 | 147.2 | 91 | 8.0 | 39 |
| CSWin-T [66] | 82.7 | - | - | 4.3 | 23 |
| Swin-S [186] | 83.0 | - | - | 8.7 | 50 |
| ViT-L | 81.5 | 349.3 | 26 | 61.6 | 305 |
| Rev-ViT-L | 81.4 | **22.6 $\downarrow$15.5$\times$** | **341 $\uparrow$13.1$\times$** | 61.6 | 305 |
| MViT-B-16 [71] | 82.8 | 153.6 | 89 | 7.8 | 37 |
| Rev-MViT-B-16 | 82.5 | **66.8 $\downarrow$2.3$\times$** | **157 $\uparrow$1.8$\times$** | 8.7 | 39 |

Table 5.1: **Comparison to prior work on ImageNet-1K classification**. All memory and maximum batch size are on 224$\times$224 input resolution on a 16G V100 GPU. **Rev-ViT** and **Rev-MViT** match performance across different FLOP regimes at a fraction of the per-input GPU memory cost.

## Video Classification

**Settings.** We also benchmark Rev-MViT-B models on action classification on Kinetics-400 [143] and Kinetics-600 [34] datasets. All the models are trained from scratch with training recipes adapted from [71].

**Results.** Table 5.2 and 5.3 present the results on action recognition task on the Kinetics-400 [143] and Kinetics-600 [34] respectively. For action recognition, we benchmark our adapted Reversible MViT model and report top-1 and top-5 performance for both datasets. Similar to the image classification benchmark, we observe that the Reversible MViT models closely match the accuracy for their non reversible counterparts at a fraction of the memory cost.

**Increasing video model batch sizes.** We note that our adapted Reversible MViT forms a very competitive video recognition model. Specifically, for both Kinetics-400 (Table 5.2) and Kinetics-600 (Table 5.3) the reversible models match the overall accuracy very closely at only **51.5**% and **37.2**% of the memory cost respectively.

This allows a batch size increase of **2**$\times$ on the 16 layer, 70.5 GFLOPs Kinetics-400 MViT-B-16 model and of **3.5**$\times$ on the 24 layer, 236 GFLOPs Kinetics-600 MViT-B-24 model, a very beneficial result for large video models which are often severely memory limited and trained with very small batch sizes (Table 5.3). Moreover, due to the more efficient design of stage-transition blocks in

| model | top-1 | **Mem** (GB) | **Max BS** | GFLOPs× views | Param |
|---|---|---|---|---|---|
| Two-Stream I3D [34] | 71.6 | - | - | 216 × NA | 25.0 |
| R(2+1)D [265] | 72.0 | - | - | 152×115 | 63.6 |
| Two-Stream R(2+1)D [265] | 73.9 | - | - | 304 × 115 | 127.2 |
| Oct-I3D + NL [46] | 75.7 | - | - | 28.9×3×10 | 33.6 |
| ip-CSN-152 [266] | 77.8 | - | - | 109×3×10 | 32.8 |
| SlowFast 4×16, R50 [77] | 75.6 | - | - | 36.1 × 30 | 34.4 |
| SlowFast 8×8, R101 [77] | 77.9 | - | - | 106 × 30 | 53.7 |
| SlowFast 8×8 +NL [77] | 78.7 | - | - | 116×3×10 | 59.9 |
| ViT-B-VTN-IN-1K [204] | 75.6 | - | - | 4218×1×1 | 114.0 |
| ViT-B-VTN-IN-21K [204] | 78.6 | - | - | 4218×1×1 | 114.0 |
| MViT-B-16 , 16×4 | 78.4 | 1.27 | 10 | 70.5×1×5 | 36.6 |
| **Rev-MViT**-B-16, 16×4 | 78.5 | **0.64** | **20** | 64×1×5 | 34.9 |

Table 5.2: **Comparison to prior work on Kinetics-400 video classification**. Single view inference cost is reported along with used number of views (FLOPs×$\text{view}_{\text{space}}$×$\text{view}_{\text{time}}$). Memory (Mem) reported in Gigabytes per input clip. Maximum Batch Size (Max BS) measured as the maximum possible single GPU batch size. All measurements are performed on a single 16G V100 GPU.

| model | top-1 | **Mem** (GB) | **Max BS** | GFLOPs× views | Param |
|---|---|---|---|---|---|
| SlowFast 16×8 +NL [77] | 81.8 | - | - | 234×3×10 | 59.9 |
| X3D-XL | 81.9 | - | - | 48.4×3×10 | 11.0 |
| ViT-B-TimeSformer-IN-21K [22] | 82.4 | - | - | 1703×3×1 | 121.4 |
| ViT-L-ViViT-IN-21K [9] | 83.0 | - | - | 3992×3×4 | 310.8 |
| MViT-B-16, 16×4 | 81.3 | - | - | 70.3×1×5 | 36.6 |
| MViT-B-16, 32×3 | 83.4 | - | - | 170×1×5 | 36.8 |
| MViT-B-24, 32×3 | 83.8 | 4.40 | 2 | 236×1×5 | 52.9 |
| **Rev-MViT**-B-24, 32×3 | 83.7 | **1.64** | **7** | 223×1×5 | 51.8 |

Table 5.3: **Comparison to prior work on Kinetics-600 video classification**. Results under same settings as Kinetics-400 in Table 5.2.

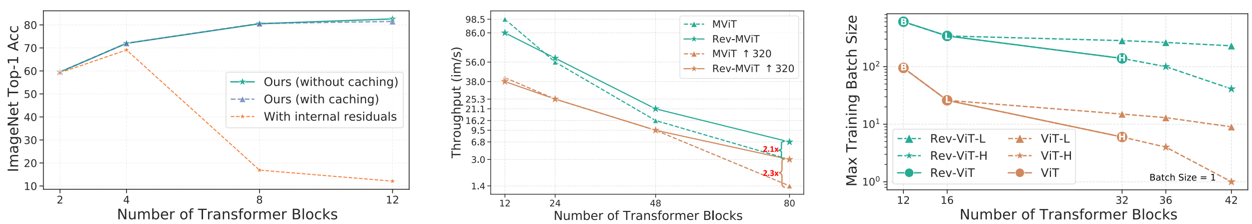
Rev-MViT (Sec. 5.3), *i.e*. bringing the dimension upsampling operation inside the pooling attention instead of being performed in the prior MLP stage [71], the Rev-MViT are also slightly more parameter and FLOP efficient on both Kinetics.

## Object Detection

We benchmark the proposed Rev-ViT-B and Rev-MViT-B models on object detection on MS-COCO [181] as well. All the models are trained on 118K training images and evaluated on the 5K validation images. We take the ViT-B and MViT-B backbones pre-trained on IN and use the

| Model | $AP^{box}$ | $AP^{mask}$ | **Memory**(GB) | GFLOPs | Param (M) |
|---|---|---|---|---|---|
| Res50 [111] | 41.0 | 37.1 | - | 260 | 44 |
| Res101 [111] | 42.8 | 38.5 | - | 336 | 63 |
| X101-64 [292] | 44.4 | 39.7 | - | 493 | 101 |
| PVT-L[275] | 44.5 | 40.7 | - | 364 | 81 |
| MViT-B | 48.2 | 43.9 | 18.9 | 668 | 57 |
| **Rev-MViT**-B | 48.0 | 43.5 | 10.9 | 683 | 58 |

Table 5.4: **Comparison on MS-COCO object detection.** Rev-MViT achieves competitive performance to MViT across all metrics at **1.7**× lower memory footprint.




(a) Activation caching and internal residuals.

(b) Training throughput vs. Model Depth

(c) Reversible training and maximum batch size.

Figure 5.3: **Ablation Experiments.** (a): We observe that (1) Learning without activation caching does not hurt reversible accuracy for Rev-ViT-B of varying depths and (2) Internal residual connections train well for shallow models but the training *diverges* for deeper models. (b): Rev-MViT has higher *throughput* for higher input resolution and deeper MViT models increasing up to **2.3**× at 224 resolution for 80 layers. (c): We benchmark the maximum batch size for Rev-ViT Base (B), Large (L) and Huge (H) and their non-reversible counterparts.

standard Mask R-CNN [114] as the detection framework. All models are trained with a standard 3× schedule (36 epochs). For MViT, we integrate the multi-scale backbone with the feature pyramid network [180]. Referring to Table 5.4 we observe that, the Rev-MViT-B model closely matches the AP performance on MViT-B at only 54.8% of the memory cost.

## Ablations

**Stronger Inherent Regularization.** Across different models and datasets, we find that at the same FLOP and parameter specifications, the reversible models tend to have stronger inherent regularization than their non-reversible counterparts. Hence, training recipes for reversible vision transformers have lighter repeated augmentations, smaller augmentation magnitudes and consequently, higher weight decay. Table 5.5 shows the effects of these recipe changes on Rev-ViT-B. We also observe

| Training Improvement | Train Acc | Top-1 ImageNet Acc |
|---|---|---|
| Naïve Rev-ViT-B | 15.3 | 12.1 |
| + Re-configuring residual streams | 82.1 | 77.2 |
| + Repeated Augmentation | 84.9 | 80.6 |
| + Lighter Augmentation magnitude | 93.2 | 81.0 |
| + Stronger Stochastic Depth | 92.0 | 81.4 |
| + Higher weight decay | 91.0 | 81.8 |
| **Rev-ViT**-B | 91.0 | 81.8 |

Table 5.5: **Rev-ViT-B Training Recipe.** We observe that reversible transformers tend to have a stronger inherent regularization and require a lighter augmented training recipe for peak performance.

| Stage-Transition Fusion | Termination Fusion | Train Acc | Top-1 Acc |
|---|---|---|---|
| Max | Norm → Concat | 78.1 | 81.7 |
| Concat | Norm → Concat | 79.1 | 82.0 |
| 2×-MLP | Norm → 2×-MLP | 80.2 | 81.8 |
| 2×-MLP + 0.2 dp | Norm → 2×-MLP → 0.5dp | 77.1 | 81.2 |
| 2×-MLP | Norm → 1-layer | 53.6 | 82.1 |
| 2×-MLP | Norm → 1-layer → 0.2dp | 64.0 | 82.4 |
| Norm → 2×-MLP | Norm → Concat | 79.4 | 82.3 |
| Norm → 2×-MLP | Norm → 1-layer → 0.2dp → Norm | 78.3 | 82.3 |
| 4×-MLP | Norm → Concat | 80.4 | 82.3 |
| 2×-MLP | Concat →Norm | 80.5 | 82.2 |
| 2×-MLP | Norm → Concat | 80.1 | 82.5 |

Table 5.6: **Lateral Fusion Strategies**. Residual streams $I_1$ and $I_2$ are fused in state-transition blocks (Sec. 5.3) as well as on termination (Sec. 5.3) before the network head. We find fusion strategy to play a key role for ReV-MViT performance. Rev-MViT-B uses a 2-layer MLP with 2× hidden dimensions in stage-transition blocks (gray). Please see Section 5.4 for details.

similar effects on other Rev-ViT and Rev-MViT models where a modified training recipe with lighter augmentations and higher weight decay play a crucial role in matching performance.

**Lateral Fusion Strategies.** The stage-transition blocks employ residual stream fusion blocks for mixing information between $I_1$ and $I_2$ (Sec. 5.3). We explore several fusion strategies in Table 5.6 using a combination of: (A) $n$×**-MLP**: Two layer perceptrons with $n$ times the hidden dimension and GELU activations. (B) $0.n$ **dp**: $n \times 10$ percent dropout on output activations. (C) Simple operators such as channel-wise maximum of $I_1$ and $I_2$ activations, and channel-wise concatenation of tensor.

Lateral connections in stage-transition stages allows effective information mixing between the residual streams and hence increases network capacity. For example, compared to concatenation, 2×-MLP increases to training accuracy by $1\%$ and also the top-1 performance by $0.5\%$. However an even heavier strategy, such as 4×-MLP widens the generalization gap and promotes over-fitting. Note that the training accuracy is often lower than the top-1 performance because of training data augmentations.

**Re-configuring residual connections.** As discussed in Sec. 5.3, the reversible vision transformer design removes the skip connections that are commonly used inside the Attention and MLP blocks (Figure 5.2). Specifically, for all of the reversible blocks in Rev-MViT and Rev-ViT, the inputs $I_1$ and $I_2$ do not have residual connections that allow residual signal propagation by directly skipping their respective functions (MLP for $I_2$ and Attention for $I_1$). Instead their residuals are performed via the other residual stream operating through the reversible transform $T$ (Sec. 5.3).

We ablate this design choice for the ViT architecture in Figure 5.3a. We vary the model depth without changing any other model dimensions and observe the performance of the two reversible variants *i.e.*, with and without internal skip connections. We note that while the naïve version with internal skip connections trains stably for shallow models, for deeper models the accuracy drops significantly. On the other hand, Rev-MViT scales well with depth, just as accurate as the other variant at shallower depths but significantly better with deeper models.

**Effect of learning without caching activations.** Figure 5.3a also compares the image classification performance of the Rev-ViT-B architecture trained with and without caching activations. This allows us to disentangle the effect of the proposed residual configurations necessary for reversible vision transformer from any artefacts that might result from learning without caching activations. However, for all depths we notice the Rev-ViT-B performance trained without caching activations to closely track the performance of the same architecture trained with caching. The slight difference at depth 12 results might stem from the training recipe being adapted for the actual Rev-ViT-B architecture trained without activations.

**Model size and Input Resolution *vs*. Throughput.** Figure 5.3b shows the training throughput comparisons for different models sizes at $224$ and $320$ input resolutions. We note that while for smaller models such as, MViT-B with a depth 12 layers, Rev-MViT-B has a slightly smaller training throughput (98.5 vs. 86.0), the additional re-computation burden of intermediate activations is easily overcome at both higher resolution training as well as for deeper models. In particular, at 224 resolution, the 24-layer and 48-layer Rev-MViT models have similar throughput as the MViT models increasing upto to **2.1**× higher throughput at 384 resolution and **2.3**× higher throughput at 384 resolution for the 80 layer depth models. Further, the rate of memory increase for deeper models is much lower for reversible variants than vanilla networks, allowing scaling to much deeper models without any additional training infrastructure burden or memory requirement like with gradient checkpointing or model parallelism.

**Maximum batch-size.** We benchmark the maximum possible batch size for Rev-ViT Base (B), Large (L) and Huge (H) and their non-reversible counterparts in Fig.5.3c. We extrapolate the trend (denoted by ) to larger models by scaling ViT-L and ViT-H in depth (keeping other model dimensions constant) and benchmark the maximum batch size for their reversible counterparts.

**Model size vs. GPU memory footprint.** Figure 5.1 plots the GPU Memory footprint for both Rev-ViT and Rev-MViT family of models as well as for several other prior networks such as MViT [71], ViT [68], ResNets and RegNetY [221]. We note that at fixed GFLOPs, reversible variants are extremely memory efficient going upto **4.5**$\times$ for MViT and **15.5**$\times$ for ViT surpassing prior convolutional variants by orders of magnitude.

## 5.5 Conclusion

We present Reversible Vision Transformers, memory-efficient reversible architectural adaptations of ViT and MViT models. We benchmark across several tasks, such as image classification, object detection and video classification and across several metrics, such as model complexity, throughput, accuracy and memory usage. Given any specification, our Rev-ViT and Rev-MViT match the accuracy of non-reversible variants at a tiny fraction of the memory cost while maintaining similar training throughput for smaller models and up to 2.3$\times$ higher throughput for larger models. Specifically, we observe that the Rev-ViT and the Rev-MViT models achieve upto 15.5$\times$ and 4.5$\times$ lighter memory footprint than ViT and MViT models respectively.

## Follow-Up Work

Chen *et al*. [318] apply the proposed Reversible Vision Transformer architecture design for Reversible Swin Transformers and apply the model for memory-efficient temporal action localization. Temporal action localization involves detecting precise temporal frame positions for the start and end boundaries of an action and hence needs to be performed at on a densely sampled video. Dense frame sampling causes GPU memory overheads during training that prohibit finetuning the backbone end-to-end on the temporal action localization task (TAL). Reversible backbone alleviate the memory overhead and allow efficient end-to-end TAL training, thereby providing significant localization performance boost.

Concurrently, [327] proposes a training procedure for reversible transformers that allows speeding up training while ensuring exact replication of the original computation. In particular, [327] proposes to stagger the activation re-computation one transformer block ahead of the gradient computation using the recomputed activations of the previous block. This allows the activations to be available for gradient calculation of the next block, as soon as the previous block finishes. Hence, the gradient calculation step does not need to wait for activation recomputation, *effectively* hiding the latency of the burden of re-computation behind latency of graident calculations, thus effectively speeding up training. This requires maintaining separate cuda work-streams that process the above two steps asynchronously. Depending on the hardware and computation size, there can be significant speedups from such operator parallelization.

# Chapter 6

# EgoSchema: A Diagnostic Benchmark for Very Long-form Video Language Understanding

## 6.1 Introduction

We introduce EgoSchema, a diagnostic benchmark for assessing very long-form video-language understanding capabilities of modern multimodal systems. Understanding long natural videos requires a host of interconnected abilities such as action and scene understanding, perceiving and tracking object states, long-term visual memory, abstract reasoning, hierarchical information aggregation, and more. Shown in Fig. 6.1 is an exemplar of the curated EgoSchema dataset. Consider the visual cognitive faculties involved in answering the question: `'What is the overarching behavior of C and the man in the video?'`. First, is the spatial recognition capabilities for disambiguating the referred character `'C'` (camera wearer) and `'the man'` as well as the present objects such as `'cards'`, `'notebook'`, `deck` as so on. Next is short-term temporal recognition capabilities of understanding the atomic actions and movement of the characters such as `'playing'`, `'taking notes'`, `'shuffling'` etc. Built upon these are the capabilities for visually understanding the mental states such `'distracted'`, `'attention'` and social dynamics such as `'teaching'`, `'showing'`. Next are medium-term actions such as `'organizing the deck'` or `'keeping track'`. Finally, long-term reasoning capabilities need to be employed for abstracting the `'overarching behavior'` of the video from all the low-level signals to be able to rule out all the other wrong options and conclude option `3` to be correct. Note that even for humans, it is impossible to answer the illustrated questions with only the shown 9 uniformly sampled frames from the three-minute video (Fig. 6.1).

This chapter is based on joint work with Raiymbek Akshulakov and Jitendra Malik [193], and is presented as it appeared in the Neurips 2023 Benchmark and Dataset proceedings.

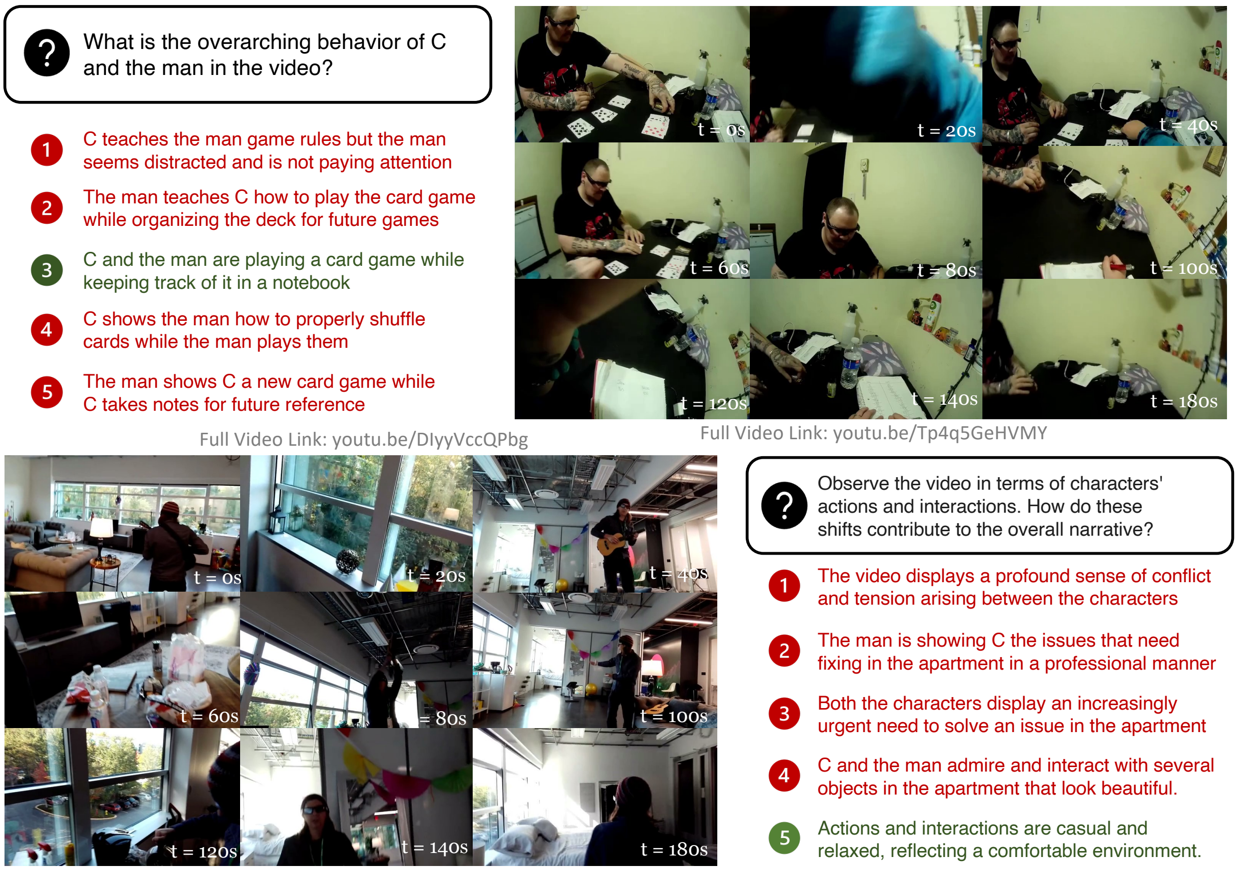



Figure 6.1: **The EgoSchema dataset** contains over 5000 very long-form video language understanding questions spanning over 250 hours of real, diverse, and high-quality egocentric video data. Each question requires choosing the correct answer out of five choices based on a *three minute* long video clip. The questions are manually curated to require very long *temporal certificates* (Sec. 6.3). EgoSchema median certificate length is about $100$ seconds, which is $5\times$ longer than the closest second dataset and $10\times$ to $100\times$ longer (Fig. 6.3) than any other video understanding dataset. State-of-the-Art video-language models consisting of billion of parameters achieve very low accuracy ($<$ 33%) in Zero-shot evaluation (random is 20%) while humans achieve about 76%. `'C'` refers to the camera wearer. Visualized clips are available at `egoschema.github.io/explorer`.

While there have been some prior attempts to formulate long-form video tasks [286, 245], they broadly tend to fall into two failure modes. The first failure mode stems from the difficulty of capturing the explosive diversity of human behavior in narrow pre-defined label spaces that leading unduly narrow and oddly specific tasks, such as like ratio or relationship prediction [286]. Hence, we propose to probe video systems capturing the rich complexity of long-form video with something just as rich and complex – natural language. However, natural language outputs are notoriously difficult to evaluate with popular metrics such as BLEU [210] and ROUGE [176] having well-known shortcomings [31]. Hence, we propose to evaluate language understanding as a multiple-choice question-answering task, thereby using the well-defined benchmark metric of overall question-answering accuracy.

The second failure mode for a long-term video task is that the proposed task happens to actually be a short-term one - only disguised as a long-term task. To measure the intrinsic "long-term" nature of a video understanding task, we propose the notion of temporal *certificate length* [10]. Intuitively, certificate length (Sec. 6.3) is the length of the video a human verifier needs to observe to be convinced of the veracity of the marked annotation. The idea of temporal certificates is not limited only to question-answering or vision-language tasks but is applicable to several video understanding tasks, including pure vision tasks such as action classification, detection, or even temporal action localization.

Based on the length of the temporal *certificate*, we propose the following temporal understanding taxonomy for video tasks: Datasets with certificate length in the order of $1$ second are termed short video tasks. Next, we name datasets with certificate length in the order of $10$ seconds as, long-form video tasks. Finally, datasets with certificate length in the order of $100$ seconds are termed as, very long-form video tasks. Fig. 6.3 presents estimates of the certificate lengths for a variety of datasets plotted against the temporal length of the video clip. We observe that the temporal certificate length is quite weakly correlated with the length of the video clip. This is due to the intentional design choice in defining the certificate set, which decouples the task of searching or retrieving the relevant sub-clip from a bigger clip from the task of visually understanding the retrieved sub-clip. And in this manner, using temporal certificate length as a metric for measuring the intrinsic temporal hardness of a dataset, avoids the failure mode of formulating an implicitly short-term task disguised as a long-term one. Section 6.3 details precise operationalizations for estimating the temporal certificate sets.

In summary, our contributions are three-fold. *First*, we propose the notion of temporal certificates, a broadly applicable notion that measures the intrinsic temporal hardness of clips in a video understanding dataset. We estimate temporal certificate lengths for a broad variety of existing datasets and show that EgoSchema has a median temporal certificate of about $100$ seconds, which is $5\times$ longer than the dataset with the second longest certificate length [286], and $25\times$ to $100\times$ longer than all other existing video understanding datasets (with or without language). *Second*, building upon the notion of temporal certificates, we introduce EgoSchema, a diagnostic benchmark for assessing the very long-form video understanding capability of multimodal video-language systems. *Third*, we benchmark both state-of-the-art video-language systems and humans in Zero-shot settings on EgoSchema to find that even the most advanced current video-language understanding systems consisting of billion of parameters achieve very low accuracy in long-from multiple-choice question-answering (< 33%) while humans achieve about $76\%$ accuracy in the unconstrained setting.

## 6.2 Related Works

**Video Question-Answering Datasets.** Visual Question-Answering [5] is a popular video-language task with several large internet-scale datasets for video-language pre-training such as Ego4D [100],

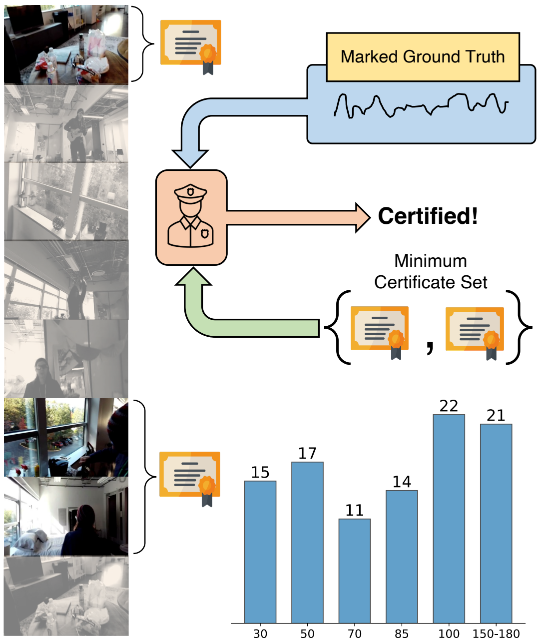



Figure 6.2: We introduce the notion of a temporal certificate set (top, Sec. 6.3), a tool to measure the intrinsic temporal length of a benchmark and show the EgoSchema certificate length distribution (bottom, Sec. 6.4) for randomly chosen 100 clips.

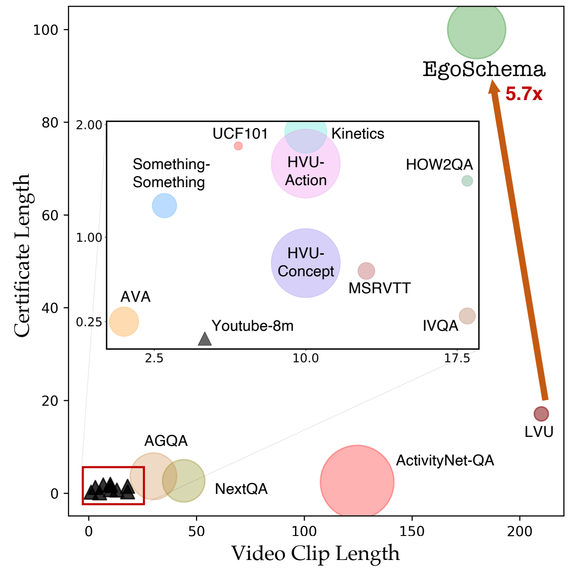



Figure 6.3: **Certificate Length across video datasets** for a broad spectrum of tasks such as action classification, detection, relationship classification, concept classification, video classification, and multiple choice question-answering. Sec. 6.4 details the precise operationalizations.

HowTo100M [200] and HowToVQA69M [182]. However, as the scope and size of pre-training datasets and models soar, it becomes critical to construct evaluations for assessing the model capabilities on various axes. Hence, many smaller datasets have been proposed for evaluating different aspects of video-language understanding such as compositional reasoning [102, 101], causal and common scene comprehension [290], instruction understanding [182, 300], video description ability [295], dynamic environments understanding [83], complex web video understanding [309], situated reasoning [285], spatiotemporal reasoning [138], social intelligence [316], dynamic neuro-symbolic reasoning [307], external knowledge-based reasoning [85] and many more [203, 306, 235, 38, 40, 74, 246, 296, 159, 160, 39, 314, 49, 163, 303, 322, 288, 107]. How2VQA69M [182] and iVQA [182] have leveraged HowTo100M [200] ASR text for generating questions. However, unlike Ego4D narrations that are used in EgoSchema, ASR text does not necessarily describe the visual elements in the scene. Hence, questions can suffer from biases where a key required information is visually absent. Additionally, generated question-answers also have quite short certificate lengths (iVQA in Fig. 6.2) due to the local nature of the ASR text.

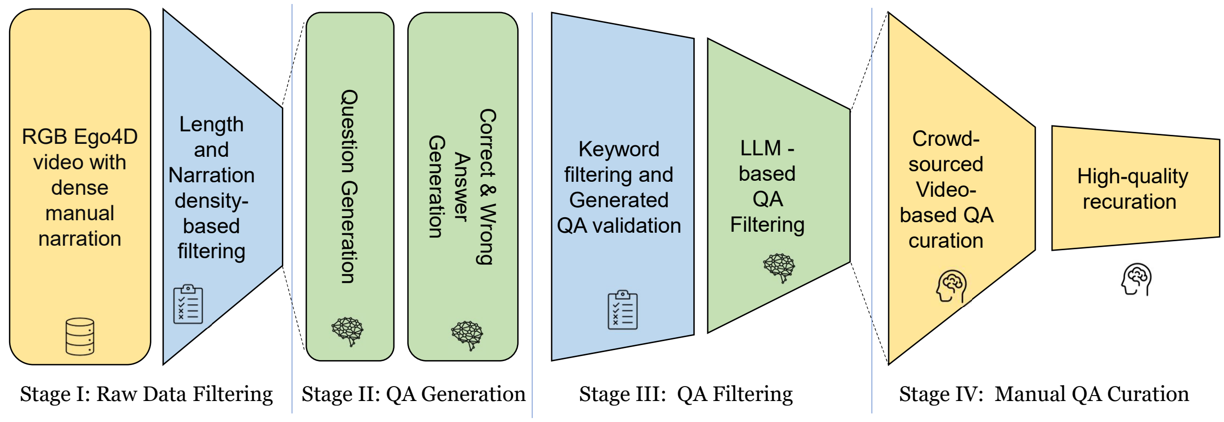



Figure 6.4: EgoSchema data pipeline. Stage I filters the suitable Ego4D RGB videos and narrations for question-answer generation (Sec. 6.3). Stage II uses narrations in a chained LLM prompting (Sec. 6.3) procedure to generate multiple $\mathcal{QAW}$ triplets per three-minute video clip (Sec. 6.3). Stage III performs pre-filtering with rule-based and LLM-based logic (Sec. 6.3). Finally, Stage IV involves two rounds of human curation on filtered $\mathcal{QAW}$ for selecting very long-form video-language understanding data (Sec. 6.3). The stage width ratios are indicative of the filter selection ratios.

**Long-form Video Understanding Datasets** have been very sparsely explored in prior works. [286] posits a long-form video understanding benchmark but the proposed tasks are unduly narrow and specific, such as the 'like' ratio and view count prediction. Also, [286] average certificate length is about $5.7\times$ smaller than EgoSchema.

[202] proposes a dataset for benchmarking efficient video inference consisting of frame-wise object mask annotations from Mask-RCNN [115] but without any long-term annotations. [240] introduces a dataset of about 111 hours of video sourced from Kinetics-400 [35] for generic event boundary detection. While the task itself requires comprehensive understanding, the video clip length is only 10 seconds long, with temporal *certificates* (Sec. 6.3) being much shorter. [256] proposes a question-answering dataset based on long movie clips but due to the open-ended nature of questions, successful approaches tend to neglect the visual data and are biased purely with approaches using additional text such as story lines. [245] proposes MAD, a language grounding dataset with an average clip of $110$ minutes. However, the length of the retrieved clip is quite short (average $4.1$ seconds) thereby resulting in a temporal *certificate* (Sec. 6.3) only a few seconds long. Further, MAD [245] and several other movie-based datasets [132, 272, 294] do not release any video data because of copyright issues. In contrast, EgoSchema has an average certificate length of about $100$ seconds. Further, EgoSchema will be publicly released under the Ego4D license, which allows direct public use of the video and text data for both research and commercial purposes.

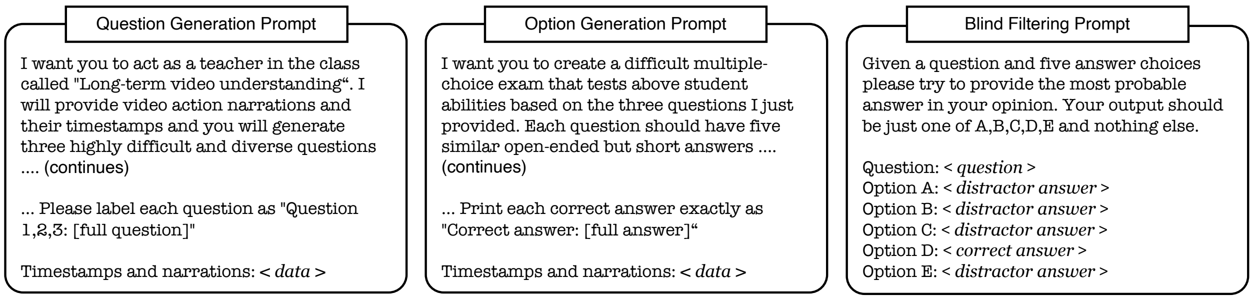



Figure 6.5: An abridged example of the generation and filtering prompts used in the EgoSchema data generation pipeline (Sec. 6.3). Full versions are provided in the *supplementary*.

## 6.3 Collecting EgoSchema

Collecting video and language datasets, even without a focus on very long-form video is quite challenging. Manually collecting, observing, and annotating videos with free-form language, in contrast to using images and pre-defined label categories, is both labor-intensive and time-consuming and thereby quite expensive. In addition to burgeoning cost, ensuring visual data diversity and minimizing visual and linguistic bias while ensuring high quality of marked annotations also contribute to the overall difficulty. All these factors get severely more challenging for long-form videos.

In this work, we propose a staged data collection pipeline (Fig. 6.4) utilizing existing large-scale but short-term video datasets, rule-based filtering procedures, and exciting new capabilities afforded by LLMs to significantly lighten the burden on human annotators. We use the proposed pipeline for curating EgoSchema, a high-quality and diverse very long-form video question-answering dataset. Associated datasheets [86] and data cards [217] for EgoSchema are provided in the *supplementary*.

### EgoSchema Pipeline

#### Stage I: Raw Data Filtering

Ego4D [100] has over 3670 hours of RGB video spread consisting of over 3.85 million narration instances covering over 1,772 unique verbs (activities) and 4,336 unique nouns (objects) [100]. The narrators are instructed to continuously pause and describe everything that the camera wearer (`'C'`) does. This creates dense and precise narrations that accurately describe the visuals.

Naturally, the collected video has non-uniform length and narration density. Since we would like to standardize the clip length for evaluation and have sufficiently rich narrations to allow interesting question-answer pairs to form in later stages, we filter the data based on the length and narration density. We choose to filter for non-overlapping three-minute clips each with at least 30 human annotated narrations (each narration is a timestamped sentence) to build EgoSchema. Detailed statistic of the number of viable clips for different possible length and narration density choices is discussed in *supplementary*.

**Stage II: Question Answer Generation**

The filtered narrations are processed with a capable LLM to generate $N$ Question-Answer triplets ($\mathcal{QAW}$), each consisting of the question $\mathcal{Q}$, the correct answer $\mathcal{A}$, and $M$ wrong answers $\mathcal{W}$, per clip. To achieve this, we experimented with several LLM inference call chaining procedures with trade-offs between quality and cost of generation that are briefly described next.

**One-shot** is the simplest prompting procedure to prompt for all $N$ instances of $\mathcal{QAW}$ in one inference call. This is the most cost-efficient option but we found the generations to be of significantly low quality. The generated $\mathcal{Q}$ often are very similar to each other and the generated $\mathcal{AW}$ have a very high false positive rate for the correct answers as well as a false negative rate for the wrong answers.
**N-shot** is the next natural prompting procedure where we generate one $\mathcal{QAW}$ per LLM inference call. This significantly improves the false positive and false negative rates but since the generated $\mathcal{Q}$ are independent and generated with the same prompt, they still tend to be very similar (comparable to one-shot), even at higher sampling temperatures. Further, the cost of generation also scales with $N$.
**QAW-shot** generates each of the $N$ questions $\mathcal{Q}$ in one inference call, followed by another inference call for generating $N$ correct answer $\mathcal{A}|\mathcal{Q}$ and finally, $N \times M$ wrong answers, $\mathcal{W}|\mathcal{Q}, \mathcal{A}$. Since each of the $N$ $\mathcal{Q}$ is generated jointly, they can be forced to be distinct with appropriate prompting. Similarly, the generated $\mathcal{A}$ and $\mathcal{W}$ can also be made distinct. However, this requires 3 *chained* LLM inference calls, and generation failures in earlier calls cascade steeply.

**Q(AW)-shot** generates each of the $N$ questions $\mathcal{Q}$ in one inference call, followed by a final inference call for generating all the $N$ correct and $N \times M$ incorrect answers in one go $\mathcal{A}, \mathcal{W}|\mathcal{Q}$. It enjoys the same uniqueness properties as QAW-shot while having just two chained calls, making it both $30\%$ cheaper and less prone to generation failure cascading. Further, between Q(AW)-shot and QAW-shot, we observe Q(AW)-shot to have a higher generated $\mathcal{A}$ quality, perhaps since LLM can jointly model $\mathcal{W}$ while generating $\mathcal{A}$. We choose this to be our main method of choice for generating $\mathcal{QAW}$.

**Prompt** for imputing narrations into the LLM has a tremendous effect on the quality of generated $\mathcal{QAW}$. We experiment with several seed prompts for each of which we inspect the quality of the $N$ generated $\mathcal{QAW}$ for $10$ clips. Based on this we iteratively improve the seed prompts manually in a zeroth order optimization fashion. In total, we experiment with a total of about $85$ prompts in this fashion to arrive at our final EgoSchema prompts – $\mathcal{P}_{\mathcal{Q}}$ for generating $N \times \mathcal{Q}$ questions and $\mathcal{P}_{\mathcal{AW}}$ for generating all remaining options $(\mathcal{AW})|\mathcal{Q}$. While we fix the $\mathcal{P}_{\mathcal{Q}}$ prompt, we use multiple $\mathcal{P}_{\mathcal{AW}}$ prompts so as to avoid any unintended bias in the options. Fig. 6.5 shows an abridged example of $\mathcal{P}_{\mathcal{Q}}$ and $\mathcal{P}_{\mathcal{AW}}$, full versions available in *supplementary* material.

**Choice of LLM** is extremely crucial for obtaining interesting long-form $\mathcal{Q}$ and generating hard negatives for $\mathcal{W}$. With weaker LLMs, the $\mathcal{Q}$ diversity across video clips remains narrow, and $\mathcal{W}$ tends to be either obviously wrong or, too similar to $\mathcal{A}$ and thus a false negative. While we experimented with both GPT-3 [26] and ChatGPT [206] but only found good quality generated $\mathcal{QAW}$ at a high enough rate with GPT-4 [207], Bard [95], and Claude [4]. For details please see *supplementary*.

We generate $N = 3$ questions per three-minute clip as well as $M = 4$ wrong answers to every question in addition to the correct answer. We observe that larger $N$ or $M$ tends to generate similar questions and wrong answers putting unnecessary pressure on Stages III and IV for filtering.

### Stage III: Generated Question Answer Filtering

While Stage II produces several high-quality $\mathcal{QAW}$, even the best LLM generations are prone to output format aberrations, hallucinations, and sometimes plain false outputs. Further, despite specific pinpointed prompts (Fig. 6.5), LLMs can fail to comply. Since, we want to ensure EgoSchema to be extremely high-quality and accurate, we set up several filtering rounds to ensure the correctness and high difficulty of questions.

**Rule-based filtering.** Keywords from the prompts such as `'long-term'`, `'narrations'`, `'timestamp'` etc. can sometimes bleed into the generated $\mathcal{QAW}$ which are then discarded. The output generations can also fail to parse according to a specified format and are also then discarded and the concerned $\mathcal{QAW}$ is regenerated.

**LLM-based filtering.** While rule-based filtering weeds out logic errors, we would like to further enrich $\mathcal{QAW}$ before employing human labor. For example, we aim to ensure EgoSchema requires grounded visual reasoning to solve, and hence questions should not be answerable *ungrounded*, without carefully observing the video. Hence, we develop a "blind" baseline.

**Blind filtering baseline** employs LLM to guess the correct answer based on the question, without having access to the video narrations conditioned on the shown filtering prompt (Fig. 6.5). All such ungrounded questions that can be answered blindly are filtered out. This also ensures that generated $\mathcal{W}$ are indeed relevant and plausible answers to $\mathcal{Q}$, since otherwise, the LLM would be able to guess $\mathcal{A}$ based only on the setting of $\mathcal{Q}$. Note that this is overly restrictive since it is possible that a question is guessed correctly through chance and is not necessarily ungrounded. However, we choose to optimize precision over recall since the amount of filtered $\mathcal{QAW}$ is still large enough.

**No-$\mathcal{Q}$ baseline.** We also experimented with a No-$\mathcal{Q}$ baseline, where the LLM is prompted to guess the correct answer using the narrations but without the question $\mathcal{Q}$. This ensures that the wrong answers are relevant and plausible to the video clip. However, we found this baseline to have near random accuracy ($\sim 20\%$), highlighting the efficacy of Stage II. Hence, we decided to not use this filter in the final pipeline. Additional details including the full prompt are in *supplementary*.

### Stage IV: Manual $\mathcal{QAW}$ Curation

While LLM filtering ensures that the generated $\mathcal{QA}$ relates to the video content, it's also necessary to ensure the veracity and a long temporal certificate length for every generated $\mathcal{QAW}$. This is achieved through a two-step manual curation process.

In the first round of curation, annotators are tasked with three primary responsibilities: **(A)** First, they verify that $\mathcal{Q}$ is well-formed and $\mathcal{A}$ is indeed the correct answer to $\mathcal{Q}$. **(B)** Next, they confirm that all the $M$ distractors, $\mathcal{W}$, are indeed wrong answers to $\mathcal{Q}$. **(C)** Finally, they ensure that the temporal certificate length for answering $\mathcal{Q}$ is at least 30 seconds.

A $\mathcal{QAW}$ is discarded if any of these three conditions are not met. This reduces the number of admissible questions by a factor of about $4\times$ to $5\times$ within the first round itself. Next is a second round of re-curation, to reinforce the conditions and guarantee data of the highest quality. We find that more than $97\%$ of the questions that pass the first round also pass the second round, speaking to the efficacy of the curation process. A crucial aspect of ensuring that the question assesses very long-form video-language understanding capabilities is the notion of temporal certificate length (condition (C) above), which we describe next. The detailed procedures for onboarding and training the human annotators, as well as the instructions for the curation process are provided in the *supplementary*.

## Temporal Certificates

We define the temporal *certificate* of a given video in a video understanding task to be the minimum set of *subclips* of the video that are both *necessary* and *sufficient* to convince a human verifier that the marked annotation for that data (such as timestamps in temporal activity localization, class label in activity recognition or, the correct option in multiple-choice question-answering) is indeed correct, without having to watch the rest of the clip outside of the certificate set (Fig. 6.2). Naturally, we define certificate length to be the sum of the temporal lengths of the sub-clips present in the certificate set.

### Meta-rules

Datasets often have implicit rules that apply uniformly across the entire dataset. We call these conventions meta-rules and allow the human verifier to be well aware of them. For example, in temporal action localization datasets [139], an implicit assumption is that the action to be localized in a contiguous sub-clip and hence can be uniquely determined by the start and end timestamps. Since this rule is valid for all data, we consider it to be a meta-rule.

A comprehensive understanding of *meta*-rules of a dataset is necessary for accurate estimation of the certificate set, and hence the certificate length. Otherwise, a spuriously long certificate might be necessary to ensure the veracity of the marked annotations. For example, consider the task of action classification on Kinetics-400. A valid meta-rule to be made available to the human verifier in this case is the mutual exclusivity of action classes i.e., each data point can belong only to one of the 400 classes present in Kinetics-400. Without this understanding, given, say a 10-second clip of a human skiing, the certificate set needs to necessarily encompass the entire 10 seconds since otherwise the human verifier might not be convinced that all of the other 399 actions are not occurring in the clip. However, with the knowledge of the label exclusivity meta-rule, the certificate length will be drastically reduced to just a fraction of a second since just observing the action of skiing in a few frames is sufficient for the human verifier to out-rule all other action classes.

Figure 6.6: **Benchmarking Zero-shot QA on EgoSchema**

| Model | Release | Inference Params | Evaluation Setting | QA Acc |
|---|---|---|---|---|
| Choosing the correct $\mathcal{A}$ uniformly at random | | | | 20.0% |
| FrozenBiLM [301] | Oct 2022 | 1.2B | 10 frames | 26.4% |
| | | | 90 frames | 26.9% |
| VIOLET [80] | Sept 2022 | 198M | 5 frames | 19.9% |
| | | | 75 frames | 19.6% |
| mPLUG-Owl [305] | May 2023 | 7.2B | 1 frame | 27.0% |
| | | | 5 frames | 31.1% |
| | | | 10 frames | 29.6% |
| | | | 15 frames | 28.7% |
| | | | 30 frames | 20.0% |
| InternVideo [279] | Dec 2022 | 478M | 10 frames | 31.4% |
| | | | 30 frames | 31.8% |
| | | | 90 frames | **32.1**% |

**Certificate Conventions**

For small certificate lengths, it is difficult for humans to estimate the exact sub-clip timestamps to be included in the certificate set. Hence, we choose to have a minimum length of $0.1$ second for a certificate. Further, in the case of two non-contiguous certificates, we collapse them into one if their closest ends are $< 5$ seconds apart. In cases where a fact needs to be verified at several places throughout the video, we let the annotator make a reasonable judgment for the length of the certificate to be included as long as it follows the above conditions.

## 6.4 Benchmarking EgoSchema

### Evaluating Certificate Lengths

Fig. 6.3 presents certificate lengths for a spectrum of tasks spread across $15$ different datasets such as, action classification (Kinetics [35], Something-Something [98], UCF101 [248], HVU-Action [62]), detection (AVA [104]), relationship classification (LVU [286]), concept classification (HVU-Concept [62]), video classification (Youtube-8M [1]), Question-Answering (NextQA [290], AGQA [101], NextQA [290], IVQA [182], MSRVTT [295], ActivityNet-QA [309], EgoSchema). For EgoSchema we benchmark the certificate length for 5 hours of video data ($100\mathcal{QAW}$) chosen randomly. For each other dataset, we ensure that (A) each annotated label class (if applicable) has at least 1 data sample evaluated and, (B) at least two hours of human effort is applied. Fig. 6.2 shows the histogram of estimated EgoSchema temporal certificate lengths for the 100 clips.

Fig. 6.3 plots the certificate length against the actual clip length. We observe that EgoSchema has temporal certificate length $5.7\times$ longer than the second longest certificate length dataset, and $10\times$ to $100\times$ longer than all other video understanding datasets.

## Evaluating Multiple-choice Question Answering on EgoSchema

In Table 6.6, We benchmark several state-of-the-art video-language models, with the intention of adding more models in the future, in a Zero-shot question-answering setting on EgoSchema. We evaluate each model in at least two settings. First is the conventional inference setting, where the model is assessed based on the same number of frames it was trained with. And second is a less challenging setting, where the model is tested on the maximum number of frames possible to execute inference with, using an 80G A100, without exceeding the GPU memory capacity. In both settings, frames are sampled uniformly from the input video clip.
**FrozenBiLM** [301] adapts frozen multi-modal encoders trained on web-scale data for the task of question answering and achieves state-of-the-art zero-shot QA accuracy across 8 video question-answering datasets. We choose the How2QA FrozenBilM model under both 10 and 90 frames.
**VIOLET** [80] a masked token modeling-based video language transformer that performs competitively on a variety of video-language tasks. We evaluate four of the best VIOLET models that are finetuned on different tasks for both 5 and 75 frames and choose the model with the best overall accuracy. More details are in *supplementary*.
**mPLUG-Owl** [305] proposes a training strategy to add image & video modality to pretrained large language models. We adapt mPLUG to facilitate the multiple choice QA by prompting the model with each of the options individually in the format: 'Given question <`question text`>, is answer <`answer text`> correct?' along with the video frames. Then, we choose the option with the highest softmax score of the token 'Yes' in the output text. We observe accuracy to be non-monotonic in frame length, and report results in 1 to 30 frames in Table 6.6.
**InternVideo** [279] proposes training video-language models jointly with masked video modeling and contrastive learning objectives. By default, InternVideo does not directly support multiple-choice video QA. We adapt the MSRVTT finetuned InternVideo model, which performs zero-shot multiple-choice tasks, by incorporating the question with each answer choice in the format: 'Question: <`question text`>? Is it <`answer text`>'. Then, we choose the option with the highest output score as the prediction. We report results spanning 10 to 90 input frames in Table 6.6. We observe that performance is monotonic with the number of frames but the gain saturates around just 30 frames.

## Human Benchmarking

We also benchmark human performance on multiple-choice question answering task on EgoSchema in Table 6.7. *First*, are time pressure settings where the annotators are asked to choose the correct answer under one (`'In <1 min'`) and three (`'In <3 min'`) minutes. Humans can already achieve an impressive 67.0% accuracy, in under 1 minute! Interestingly, this only slightly increases (+1.0%) when allowed three minutes. We believe that this can inform about performance on

EgoSchema in limited model inference capacities. We believe this could inform about the frame rate needed for long-form video understanding in future models. *Second*, we also benchmark human performance using only 1 fps video (`'180 frames'`). Surprisingly, we observe that just with 1 fps humans can achieve an impressive 67.2%.

*Third*, we evaluate human performance in a restrictive setting where the annotator is forced to first watch the video without reading the text, and then answer the question without re-watching the video (`'Video ⟶ Text'`). Curiously, this achieves better accuracy than the `'No constraint'` setting where the annotators are asked to simply answer without any constraints (76.2% vs. 75.0%). A possible hypothesis is that watching the video without text allows the annotator to focus more closely on the video, thereby benefiting performance than the setting where the attention is somewhat divided between the text and video. We believe this will help us understand the performance trade-offs in the early vs. late fusion of video and text modalities for long-form video-language models. Accuracy for `'No constraint'` setting is estimated over 9 hours of video. All other accuracies are estimated over 5 hours of video.

## 6.5 Conclusion

We present EgoSchema, a novel diagnostic benchmark designed for assessing very long-form video-language understanding capabilities of modern multimodal models. We also introduce the notion of a temporal *certificate* set, a probe that can be applied to a wide array of video tasks and benchmarks for understanding their intrinsic temporal lengths. We estimate temporal certificates of 15 varied datasets and demonstrate EgoSchema to exhibit temporal certificate length approximately $5.7\times$ longer than the next longest dataset and $25\times$ to $100\times$ longer than all other video understanding datasets. We also benchmark several state-of-the-art models on EgoSchema and find their Zero-shot question-answering accuracy to be less than $33\%$ while humans achieve 76%. We believe that EgoSchema will play a key role in the development and evaluation of future very long-form video-language models.

Figure 6.7: **Human Accuracy on EgoSchema**

| Evaluation Setting | QA Accuracy |
|---|---|
| 180 frames | 67.2% |
| In <1 min | 67.0% |
| In <3 min | 68.0% |
| No constraint | 75.0% |
| Video $\rightarrow$ Text | **76.2**% |

# Chapter 7

# Conclusion

In this thesis, I presented at a multifaceted approach to develop techniques that enable development of video models for perceiving people over long periods. This endeavor addressed the crucial gaps in contemporary computer vision systems, particularly in long-term human perception. The work was grounded on the synergy of algorithms, architectures, and datasets, with substantial contributions presented in each area.

In Chapter 2 and Chapter 3, I presented techniques in the realm of algorithms, where we tackled the challenge of long-term human motion forecasting. I presented models, PECNet and Y-net, that significantly pushed the boundaries of trajectory forecasting capabilities of the then prevalent motion forecasting models and deeply influence even the current state-of-the-art models that have adopted the presented endpoint and waypoint design. PECNet proposed a novel approach by focusing on trajectory endpoints to model long-term agent intent. Building on this, Y-net extended these concepts to scene-aware motion forecasting, enabling minute-long forecasts with enhanced interpretability through local distribution heat maps. These models represent a leap forward in understanding human motion and intent over longer time horizons and their central tenets of endpoint and waypoint conditioning are adopted in even the current state-of-the-art motion forecasting models.

In terms of architectures, we introduced Multiscale Vision Transformers (MViT) in Chapter 4 and Reversible Vision Transformers in Chapter 5. MViT brings in a new perspective to visual inputs processing in transformers, integrating hierarchical visual priors into the transformer deisng and achieves remarkable FLOP efficiency across several video tasks and with a simple extension to image tasks as well. Reversible Vision Transformers further advance vision transformers by drastically reducing the memory footprint required for training, enabling deeper and more complex transformer architectures. These architectural innovations are not only pivotal for long-term video understanding but also have broader implications across various efficiency improvements in vision-related applications.

Lastly, In Chapter 6, I presented EgoSchema, the first truly long-form video-lagunage understanding benchmark. I present the notion of temporal certificates and an LLM based generation, filtering and curation pipeline for collecting EgoSchema. EgoSchema stands as a robust tool for evaluating and advancing the state-of-the-art in long-term video understanding. The benchmarking results using EgoSchema have highligh the existing performance gap between current models and human-level understanding, setting a clear direction for future research in the area of long-form video understanding.

In conclusion, this thesis presents a holistic approach to improving the perception of people over long periods by video models. The advancements in algorithms, architectures, and datasets not only address existing limitations but also open new avenues for research and application. While we have made significant strides, long-form video understanding continues to be one of the last-standing grand challenges of computer vision and the journey towards truly achieving it is still ongoing. The insights and tools developed in this thesis lay a strong foundation for future explorations in this exciting and ever-evolving field of study.

# Bibliography

[1] Sami Abu-El-Haija et al. "Youtube-8m: A large-scale video classification benchmark". In: *arXiv preprint arXiv:1609.08675* (2016).

[2] Alexandre Alahi, Vignesh Ramanathan, and Li Fei-Fei. "Socially-aware large-scale crowd forecasting". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 2203–2210.

[3] Alexandre Alahi et al. "Social lstm: Human trajectory prediction in crowded spaces". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 961–971.

[4] anthropic. *Introducing Claude*. 2023. URL: https://www.anthropic.com/index/introducing-claude.

[5] Stanislaw Antol et al. "Vqa: Visual question answering". In: *Proc. ICCV*. 2015.

[6] Georges Aoude et al. "Mobile agent trajectory prediction using Bayesian nonparametric reachability trees". In: *Infotech@ Aerospace 2011*. 2011, p. 1512.

[7] Ian A Apperly and Stephen A Butterfill. "Do humans have two systems to track beliefs and belief-like states?" In: *Psychological review* 116.4 (2009), p. 953.

[8] Anurag Arnab et al. "ViViT: A Video Vision Transformer". In: *arXiv preprint arXiv:2103.15691* (2021).

[9] Anurag Arnab et al. "Vivit: A video vision transformer". In: *Proc. ICCV*. 2021.

[10] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[11] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. "Layer normalization". In: *arXiv preprint arXiv:1607.06450* (2016).

[12] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. "Layer normalization". In: *arXiv:1607.06450* (2016).

[13] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473* (2014).

[14] Chris L Baker, Rebecca Saxe, and Joshua B Tenenbaum. "Action understanding as inverse planning". In: *Cognition* 113.3 (2009), pp. 329–349.

[15] Lamberto Ballan et al. "Knowledge transfer for scene-specific motion prediction". In: *European Conference on Computer Vision*. Springer. 2016, pp. 697–713.

[16] Josh Beal et al. "Toward Transformer-Based Object Detection". In: *arXiv preprint arXiv:2012.09958* (2020).

[17] Jens Behrmann et al. "Invertible residual networks". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 573–582.

[18] Irwan Bello. "LambdaNetworks: Modeling Long-Range Interactions Without Attention". In: *International Conference on Learning Representations* (2021).

[19] Irwan Bello et al. "Attention Augmented Convolutional Networks". In: *International Conference on Computer Vision* (2019).

[20] Maren Bennewitz, Wolfram Burgard, and Sebastian Thrun. "Learning motion patterns of persons for mobile service robots". In: *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*. Vol. 4. IEEE. 2002, pp. 3601–3606.

[21] Maren Bennewitz et al. "Learning motion patterns of people for compliant robot motion". In: *The International Journal of Robotics Research* 24.1 (2005), pp. 31–48.

[22] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. "Is Space-Time Attention All You Need for Video Understanding?" In: *arXiv preprint arXiv:2102.05095* (2021).

[23] Apratim Bhattacharyya et al. "Conditional Flow Variational Autoencoders for Structured Sequence Prediction". In: *arXiv preprint arXiv:1908.09008* (2019).

[24] Julian Bock et al. "The inD Dataset: A Drone Dataset of Naturalistic Road User Trajectories at German Intersections". In: 2019.

[25] Andrew Brock, Jeff Donahue, and Karen Simonyan. "Large scale gan training for high fidelity natural image synthesis". In: *arXiv preprint arXiv:1809.11096* (2018).

[26] Tom Brown et al. "Language models are few-shot learners". In: *NIPS* (2020).

[27] Tom B Brown et al. "Language models are few-shot learners". In: *arXiv preprint arXiv:2005.14165* (2020).

[28] Robin Brügger, Christian F Baumgartner, and Ender Konukoglu. "A partially reversible u-net for memory-efficient volumetric image segmentation". In: *International conference on medical image computing and computer-assisted intervention*. Springer. 2019, pp. 429–437.

[29] Peter J Burt and Edward H Adelson. "The Laplacian pyramid as a compact image code". In: *Readings in computer vision*. Elsevier, 1987, pp. 671–679.

[30] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. "Coco-stuff: Thing and stuff classes in context". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1209–1218.

[31] Chris Callison-Burch, Miles Osborne, and Philipp Koehn. "Re-evaluating the role of BLEU in machine translation research". In: 2006.

[32] Zhe Cao et al. "Long-term human motion prediction with scene context". In: *Proc. ECCV*. 2020.

[33] Nicolas Carion et al. "End-to-end object detection with transformers". In: *Proc. ECCV*. Springer. 2020, pp. 213–229.

[34] Joao Carreira and Andrew Zisserman. "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset". In: *Proc. CVPR*. 2017.

[35] Joao Carreira and Andrew Zisserman. "Quo vadis, action recognition? a new model and the kinetics dataset". In: *Proc. CVPR*. 2017.

[36] Joao Carreira et al. "A Short Note about Kinetics-600". In: *arXiv:1808.01340* (2018).

[37] João Carreira et al. "A Short Note on the Kinetics-700 Human Action Dataset". In: *arXiv preprint arXiv:1907.06987* (2019).

[38] Santiago Castro et al. "FIBER: Fill-in-the-blanks as a challenging video understanding evaluation framework". In: *arXiv preprint arXiv:2104.04182* (2021).

[39] Santiago Castro et al. "LifeQA: A real-life dataset for video question answering". In: 2020.

[40] Santiago Castro et al. "WildQA: In-the-Wild Video Question Answering". In: *arXiv preprint arXiv:2209.06650* (2022).

[41] Bo Chang et al. "Reversible architectures for arbitrarily deep residual neural networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[42] Chun-Fu Chen et al. "Big-little net: An efficient multi-scale feature representation for visual and speech recognition". In: *arXiv preprint arXiv:1807.03848* (2018).

[43] Jieneng Chen et al. "Transunet: Transformers make strong encoders for medical image segmentation". In: *arXiv preprint arXiv:2102.04306* (2021).

[44] Lili Chen et al. "Decision transformer: Reinforcement learning via sequence modeling". In: *arXiv preprint arXiv:2106.01345* (2021).

[45] Mark Chen et al. "Generative pretraining from pixels". In: *Proc. ICML*. PMLR. 2020, pp. 1691–1703.

[46] Yunpeng Chen et al. "Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution". In: *arXiv preprint arXiv:1904.05049* (2019).

[47] Zhengsu Chen et al. "Visformer: The vision-friendly transformer". In: *Proc. ICCV*. 2021.

[48] Rewon Child et al. "Generating long sequences with sparse transformers". In: *arXiv preprint arXiv:1904.10509* (2019).

[49] Seongho Choi et al. "DramaQA: Character-centered video story understanding with hierarchical qa". In: 2021.

[50] Krzysztof Choromanski et al. "Rethinking attention with performers". In: *arXiv preprint arXiv:2009.14794* (2020).

[51] Xiangxiang Chu et al. "Do We Really Need Explicit Position Encodings for Vision Transformers?" In: *arXiv preprint arXiv:2102.10882* (2021).

[52] Paper with code. *ETH/UCY Trajectory Prediction Benchmark, https://paperswithcode.com/sota/trajectory-prediction-on-ethucy*. 2020. (Visited on 11/11/2020).

[53] Paper with code. *Stanford Drone Trajectory Prediction Benchmark, https://paperswithcode.com/sota/trajectory-prediction-on-stanford-drone*. 2020. (Visited on 11/11/2020).

[54] Ekin D Cubuk et al. "Randaugment: Practical automated data augmentation with a reduced search space". In: *Proc. CVPR*. 2020.

[55] Zihang Dai et al. "Funnel-transformer: Filtering out sequential redundancy for efficient language processing". In: *arXiv preprint arXiv:2006.03236* (2020).

[56] Jeffrey Dean et al. "Large scale distributed deep networks". In: *Advances in neural information processing systems* 25 (2012), pp. 1223–1231.

[57] J. Deng et al. "ImageNet: A Large-Scale Hierarchical Image Database". In: *Proc. CVPR*. 2009.

[58] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *Proc. CVPR*. Ieee. 2009, pp. 248–255.

[59] Nachiket Deo and Mohan M Trivedi. "Trajectory Forecasts in Unknown Environments Conditioned on Grid-Based Plans". In: *arXiv preprint arXiv:2001.00735* (2020).

[60] Armen Der Kiureghian and Ove Ditlevsen. "Aleatory or epistemic? Does it matter?" In: *Structural safety* 31.2 (2009), pp. 105–112.

[61] Karan Desai and Justin Johnson. "Virtex: Learning visual representations from textual annotations". In: *arXiv preprint arXiv:2006.06666* (2020).

[62] Ali Diba et al. "Large scale holistic video understanding". In: *Proc. ECCV*. 2020.

[63] Laurent Dinh, David Krueger, and Yoshua Bengio. "Nice: Non-linear independent components estimation". In: *arXiv preprint arXiv:1410.8516* (2014).

[64] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. "Density estimation using real nvp". In: *arXiv preprint arXiv:1605.08803* (2016).

[65] Piotr Dollár, Mannat Singh, and Ross Girshick. "Fast and Accurate Model Scaling". In: *Proc. CVPR*. 2021.

[66] Xiaoyi Dong et al. "Cswin transformer: A general vision transformer backbone with cross-shaped windows". In: *arXiv preprint arXiv:2107.00652* (2021).

[67] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[68] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *Proc. ICLR*. 2021.

[69] Sergey Edunov et al. “Understanding back-translation at scale”. In: *arXiv preprint arXiv:1808.09381* (2018).

[70] Christian Etmann, Rihuan Ke, and Carola-Bibiane Schönlieb. “iUNets: Fully invertible U-Nets with learnable up-and downsampling”. In: *arXiv preprint arXiv:2005.05220* (2020).

[71] Haoqi Fan et al. “Multiscale vision transformers”. In: *Proc. ICCV*. 2021.

[72] Haoqi Fan et al. *PySlowFast*. https://github.com/facebookresearch/slowfast. 2020.

[73] Haoqi Fan et al. “PyTorchVideo: A Deep Learning Library for Video Understanding”. In: *Proceedings of the 29th ACM International Conference on Multimedia*. https://pytorchvideo.org/. 2021.

[74] Zhiyuan Fang et al. “Video2commonsense: Generating commonsense descriptions to enrich video captioning”. In: *arXiv preprint arXiv:2003.05162* (2020).

[75] Christoph Feichtenhofer. “X3d: Expanding architectures for efficient video recognition”. In: *Proc. CVPR*. 2020, pp. 203–213.

[76] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. “Convolutional Two-Stream Network Fusion for Video Action Recognition”. In: *Proc. CVPR*. 2016.

[77] Christoph Feichtenhofer et al. “SlowFast networks for video recognition”. In: *Proc. ICCV*. 2019.

[78] Christoph Feichtenhofer et al. *SlowFast networks for video recognition in ActivityNet challenge 2019*. http://static.googleusercontent.com/media/research.google.com/en//ava/2019/fair_slowfast.pdf. 2019.

[79] Marc Finzi et al. “Invertible convolutional networks”. In: *Workshop on Invertible Neural Nets and Normalizing Flows, International Conference on Machine Learning*. 2019.

[80] Tsu-Jui Fu et al. “An empirical study of end-to-end video-language transformers with masked visual modeling”. In: *arXiv preprint arXiv:2209.01540* (2022).

[81] Kunihiko Fukushima and Sei Miyake. “Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition”. In: *Competition and cooperation in neural nets*. Springer, 1982, pp. 267–285.

[82] Valentin Gabeur et al. “Multi-modal transformer for video retrieval”. In: *Proc. ECCV*. Vol. 5. Springer. 2020.

[83] Difei Gao et al. “Env-QA: a video question answering benchmark for comprehensive understanding of dynamic environments”. In: *Proc. ICCV*. 2021.

[84] Shanghua Gao et al. “Res2net: A new multi-scale backbone architecture”. In: *IEEE PAMI* (2019).

[85] Noa Garcia et al. “KnowIT VQA: Answering knowledge-based questions about videos”. In: 2020.

[86] Timnit Gebru et al. “Datasheets for datasets”. In: *Communications of the ACM* (2021).

[87] Amir Gholami et al. “Ai and memory wall”. In: *RiseLab Medium Post* (2021).

[88] Rohit Girdhar et al. “Video action transformer network”. In: *Proc. CVPR*. 2019.

[89] Ross Girshick. “Fast R-CNN”. In: *Proc. ICCV*. 2015.

[90] Michael Goldhammer et al. “Pedestrian’s trajectory forecast in public traffic with artificial neural networks”. In: *2014 22nd International Conference on Pattern Recognition*. IEEE. 2014, pp. 4110–4115.

[91] Aidan N Gomez et al. “The reversible residual network: Backpropagation without storing activations”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017, pp. 2211–2221.

[92] Aidan N Gomez et al. “The reversible residual network: Backpropagation without storing activations”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017, pp. 2211–2221.

[93] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. “6.2. 2.3 softmax units for multinoulli output distributions”. In: *Deep Learning*. MIT Press, 2016, pp. 180–184.

[94] Ian Goodfellow et al. “Generative adversarial nets”. In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[95] Google. *An important next step on our AI journey*. 2023. URL: https://blog.google/technology/ai/bard-google-ai-search-updates/.

[96] Priya Goyal et al. “Accurate, large minibatch SGD: training ImageNet in 1 hour”. In: *arXiv:1706.02677* (2017).

[97] Raghav Goyal et al. “The “Something Something" Video Database for Learning and Evaluating Visual Common Sense.” In: *ICCV*. 2017.

[98] Raghav Goyal et al. “The" something something" video database for learning and evaluating visual common sense”. In: *Proc. CVPR*. 2017.

[99] Benjamin Graham et al. “LeViT: A Vision Transformer in ConvNet’s Clothing for Faster Inference”. In: *Proc. ICCV*. 2021.

[100] Kristen Grauman et al. “Ego4d: Around the world in 3,000 hours of egocentric video”. In: *Proc. CVPR*. 2022.

[101] Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. “AGQA 2.0: An updated benchmark for compositional spatio-temporal reasoning”. In: *arXiv preprint arXiv:2204.06105* (2022).

[102] Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. “Agqa: A benchmark for compositional spatio-temporal reasoning”. In: *Proc. CVPR*. 2021.

[103] Chunhui Gu et al. “AVA: A Video Dataset of Spatio-Temporally Localized Atomic Visual Actions”. In: *Proc. CVPR*. 2018.

[104] Chunhui Gu et al. “Ava: A video dataset of spatio-temporally localized atomic visual actions”. In: *Proc. CVPR*. 2018.

[105] Meng-Hao Guo et al. "PCT: Point Cloud Transformer". In: *arXiv preprint arXiv:2012.09688* (2020).

[106] Agrim Gupta et al. "Social gan: Socially acceptable trajectories with generative adversarial networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2255–2264.

[107] Pranay Gupta and Manish Gupta. "NEWSKVQA: Knowledge-Aware News Video Question Answering". In: *Proc. KDD*. Springer.

[108] Zhang Hang et al. "Resnest: Split-attention networks". In: 2020.

[109] Boris Hanin and David Rolnick. "How to start training: The effect of initialization and architecture". In: *arXiv preprint arXiv:1803.01719* (2018).

[110] Tristan Hascoet et al. "Layer-Wise Invertibility for Extreme Memory Cost Reduction of CNN Training". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 2019, pp. 0–0.

[111] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *Proc. CVPR*. 2016.

[112] Kaiming He et al. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proc. CVPR*. 2015.

[113] Kaiming He et al. "Identity Mappings in Deep Residual Networks". In: *Proc. ECCV*. 2016.

[114] Kaiming He et al. "Mask R-CNN". In: *Proc. ICCV*. 2017.

[115] Kaiming He et al. "Mask r-cnn". In: *Proc. CVPR*. 2017.

[116] Shuting He et al. "TransReID: Transformer-based Object Re-Identification". In: *arXiv preprint arXiv:2102.04378* (2021).

[117] Dirk Helbing. *Stochastische Methoden, nichtlineare Dynamik und quantitative Modelle sozialer Prozesse*. Shaker, 1993.

[118] Dirk Helbing and Peter Molnar. "Social force model for pedestrian dynamics". In: *Physical review E* 51.5 (1995), p. 4282.

[119] Dan Hendrycks and Kevin Gimpel. "Gaussian error linear units (GELUs)". In: *arXiv preprint arXiv:1606.08415* (2016).

[120] Geoffrey E Hinton et al. "Improving neural networks by preventing co-adaptation of feature detectors". In: *arXiv:1207.0580* (2012).

[121] Jonathan Ho and Stefano Ermon. "Generative adversarial imitation learning". In: *Advances in neural information processing systems*. 2016, pp. 4565–4573.

[122] Jonathan Ho et al. "Flow++: Improving flow-based generative models with variational dequantization and architecture design". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2722–2730.

[123] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[124] Elad Hoffer et al. "Augment Your Batch: Improving Generalization Through Instance Repetition". In: *Proc. CVPR*. 2020, pp. 8129–8138.

[125] Mark Horowitz. "1.1 computing's energy problem (and what we can do about it)". In: *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. IEEE. 2014, pp. 10–14.

[126] Han Hu et al. "Local relation networks for image recognition". In: *Proc. ICCV*. 2019, pp. 3464–3473.

[127] Han Hu et al. "Relation networks for object detection". In: *Proc. CVPR*. 2018, pp. 3588–3597.

[128] Ronghang Hu and Amanpreet Singh. "Transformer is All You Need: Multimodal Multitask Learning with a Unified Transformer". In: *arXiv preprint arXiv:2102.10772* (2021).

[129] Cheng-Zhi Anna Huang et al. "Music transformer". In: *ICLR*. 2019.

[130] Gao Huang et al. "Deep networks with stochastic depth". In: *Proc. ECCV*. 2016.

[131] Jun-Jie Huang and Pier Luigi Dragotti. "WINNet: Wavelet-inspired Invertible Network for Image Denoising". In: *arXiv preprint arXiv:2109.06381* (2021).

[132] Qingqiu Huang et al. "MovieNet: A Holistic Dataset for Movie Understanding". In: *Proc. ECCV*. 2020.

[133] DH Hubel and TN Wiesel. "Receptive fields of optic nerve fibres in the spider monkey". In: *The Journal of physiology* 154.3 (1960), pp. 572–580.

[134] Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. "Timeception for Complex Action Recognition". In: *Proc. CVPR*. 2019.

[135] Andrei Ivanov et al. "Data movement is all you need: A case study on optimizing transformers". In: *arXiv preprint arXiv:2007.00072* (2020).

[136] Boris Ivanovic and Marco Pavone. "The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 2375–2384.

[137] Andrew Jaegle et al. "Perceiver: General perception with iterative attention". In: *arXiv preprint arXiv:2103.03206* (2021).

[138] Yunseok Jang et al. "Tgif-qa: Toward spatio-temporal reasoning in visual question answering". In: *Proc. CVPR*. 2017.

[139] Yu-Gang Jiang et al. *THUMOS challenge: action recognition with a large number of classes (2014)*. 2014. URL: https://www.crcv.ucf.edu/THUMOS14/.

[140] Huaizu Jiang et al. "Super slomo: High quality estimation of multiple intermediate frames for video interpolation". In: *Proc. CVPR*. 2018.

[141] Eugen Käfer et al. "Recognition of situation classes at road intersections". In: *2010 IEEE International Conference on Robotics and Automation*. IEEE. 2010, pp. 3960–3965.

[142] Daniel Kahneman. *Thinking, fast and slow*. Macmillan, 2011.

[143] Will Kay et al. "The kinetics human action video dataset". In: *arXiv:1705.06950* (2017).

[144] Christoph G Keller and Dariu M Gavrila. "Will the pedestrian cross? a study on pedestrian path prediction". In: *IEEE Transactions on Intelligent Transportation Systems* 15.2 (2013), pp. 494–506.

[145] ByeoungDo Kim et al. "Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network". In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2017, pp. 399–404.

[146] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[147] Diederik P Kingma and Prafulla Dhariwal. "Glow: Generative flow with invertible 1x1 convolutions". In: *arXiv preprint arXiv:1807.03039* (2018).

[148] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[149] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. "Reformer: The efficient transformer". In: *arXiv preprint arXiv:2001.04451* (2020).

[150] Kris M Kitani et al. "Activity forecasting". In: *European Conference on Computer Vision*. Springer. 2012, pp. 201–214.

[151] Jan J Koenderink. "The structure of images". In: *Biological cybernetics* 50.5 (1984), pp. 363–370.

[152] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *NIPS*. 2012.

[153] Eckhard Kruse and Friedrich M Wahl. "Camera-based observation of obstacle motions to derive statistical data for mobile robot motion planning". In: *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*. Vol. 1. IEEE. 1998, pp. 662–667.

[154] Tomasz Piotr Kucner et al. "Enabling flow awareness for mobile robots in partially observable environments". In: *IEEE Robotics and Automation Letters* 2.2 (2017), pp. 1093–1100.

[155] Heeseung Kwon et al. "Motionsqueeze: Neural motion feature learning for video understanding". In: *Proc. ECCV*. Springer. 2020, pp. 345–362.

[156] Hung Le et al. "Multimodal transformer networks for end-to-end video-grounded dialogue systems". In: *arXiv preprint arXiv:1907.01166* (2019).

[157] Yann LeCun et al. "Backpropagation applied to handwritten zip code recognition". In: *Neural computation* 1.4 (1989), pp. 541–551.

[158] Namhoon Lee et al. “DESIRE: Distant future prediction in dynamic scenes with interacting agents”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 336–345.

[159] Jie Lei et al. “Tvqa: Localized, compositional video question answering”. In: *arXiv preprint arXiv:1809.01696* (2018).

[160] Jie Lei et al. “Tvqa+: Spatio-temporal grounding for video question answering”. In: *arXiv preprint arXiv:1904.11574* (2019).

[161] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. “Crowds by example”. In: *Computer graphics forum*. Vol. 26. 3. Wiley Online Library. 2007, pp. 655–664.

[162] Duo Li and Shang-Hua Gao. “m-RevNet: Deep Reversible Neural Networks with Momentum”. In: *arXiv preprint arXiv:2108.05862* (2021).

[163] Guangyao Li et al. “Learning to answer questions in dynamic audio-visual scenarios”. In: *Proc. CVPR*. 2022.

[164] Guohao Li et al. “Training Graph Neural Networks with 1000 Layers”. In: *arXiv preprint arXiv:2106.07476* (2021).

[165] Jiachen Li, Hengbo Ma, and Masayoshi Tomizuka. “Conditional generative neural system for probabilistic trajectory prediction”. In: *arXiv preprint arXiv:1905.01631* (2019).

[166] Liunian Harold Li et al. “Visualbert: A simple and performant baseline for vision and language”. In: *arXiv preprint arXiv:1908.03557* (2019).

[167] Naihan Li et al. “Neural speech synthesis with transformer network”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 6706–6713.

[168] Rui Li, Chenxi Duan, and Shunyi Zheng. “Linear Attention Mechanism: An Efficient Attention for Semantic Segmentation”. In: *arXiv preprint arXiv:2007.14902* (2020).

[169] Shanshan Li et al. “Attention-Aware Invertible Hashing Network”. In: *International Conference on Image and Graphics*. Springer. 2019, pp. 409–420.

[170] Shaohui Li et al. “REV-AE: A learned frame set for image reconstruction”. In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 1823–1827.

[171] Yan Li et al. “Tea: Temporal excitation and aggregation for action recognition”. In: *Proc. CVPR*. 2020, pp. 909–918.

[172] Zhenyang Li et al. “VideoLSTM convolves, attends and flows for action recognition”. In: *Computer Vision and Image Understanding* 166 (2018), pp. 41–50.

[173] Junwei Liang, Lu Jiang, and Alexander Hauptmann. “SimAug: Learning Robust Representations from 3D Simulation for Pedestrian Trajectory Prediction in Unseen Cameras”. In: (2020).

[174] Junwei Liang et al. *The Garden of Forking Paths: Towards Multi-Future Trajectory Prediction*. 2020. arXiv: 1912.06445 [cs.CV].

[175] Lin Liao et al. “Voronoi tracking: Location estimation using sparse and noisy sensor data”. In: *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*. Vol. 1. IEEE. 2003, pp. 723–728.

[176] Chin-Yew Lin. “Rouge: A package for automatic evaluation of summaries”. In: 2004.

[177] Ji Lin, Chuang Gan, and Song Han. “Temporal shift module for efficient video understanding”. In: *Proc. ICCV*. 2019.

[178] Ji Lin, Chuang Gan, and Song Han. “Tsm: Temporal shift module for efficient video understanding”. In: *Proc. CVPR*. 2019, pp. 7083–7093.

[179] Kevin Lin, Lijuan Wang, and Zicheng Liu. “End-to-End Human Pose and Mesh Reconstruction with Transformers”. In: *arXiv preprint arXiv:2012.09760* (2020).

[180] Tsung-Yi Lin et al. “Feature pyramid networks for object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125.

[181] Tsung-Yi Lin et al. “Microsoft COCO: Common objects in context”. In: *Proc. ECCV*. 2014.

[182] Feng Liu et al. “ivqa: Inverse visual question answering”. In: *Proc. CVPR*. 2018.

[183] Kang Liu et al. “Semantics-to-Signal Scalable Image Compression with Learned Revertible Representations”. In: *International Journal of Computer Vision* (2021), pp. 1–17.

[184] Wei Liu et al. “CPTR: Full Transformer Network for Image Captioning”. In: *arXiv preprint arXiv:2101.10804* (2021).

[185] Yang Liu et al. “Invertible Denoising Network: A Light Solution for Real Noise Removal”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 13365–13374.

[186] Ze Liu et al. “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows”. In: *arXiv preprint arXiv:2103.14030* (2021).

[187] Ze Liu et al. “Swin transformer: Hierarchical vision transformer using shifted windows”. In: *Proc. CVPR*. 2022.

[188] Francesco Locatello et al. “Object-centric learning with slot attention”. In: *arXiv preprint arXiv:2006.15055* (2020).

[189] Ilya Loshchilov and Frank Hutter. “Fixing weight decay regularization in adam”. In: (2018).

[190] Ilya Loshchilov and Frank Hutter. “SGDR: Stochastic gradient descent with warm restarts”. In: *arXiv:1608.03983* (2016).

[191] Jiasen Lu et al. “Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks”. In: *arXiv preprint arXiv:1908.02265* (2019).

[192] Matthew MacKay et al. “Reversible recurrent neural networks”. In: *arXiv preprint arXiv:1810.10999* (2018).

[193] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. “Egoschema: A diagnostic benchmark for very long-form video language understanding”. In: *arXiv preprint arXiv:2308.09126* (2023).

[194] Karttikeya Mangalam et al. *Disentangling Human Dynamics for Pedestrian Locomotion Forecasting with Noisy Supervision*. 2020. arXiv: 1911.01138 [cs.CV].

[195] Karttikeya Mangalam et al. "From goals, waypoints & paths to long term human trajectory forecasting". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 15233–15242.

[196] Karttikeya Mangalam et al. "It Is Not the Journey but the Destination: Endpoint Conditioned Trajectory Prediction". In: *arXiv preprint arXiv:2004.02025* (2020).

[197] Karttikeya Mangalam et al. "Reversible vision transformers". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 10830–10840.

[198] A. A. Markov. "An Example of Statistical Investigation of the Text Eugene Onegin Concerning the Connection of Samples in Chains". In: *Lecture at the physical-mathematical faculty, Royal Academy of Sciences, St. Petersburg* (23 January 1913).

[199] Ramin Mehran, Alexis Oyama, and Mubarak Shah. "Abnormal crowd behavior detection using social force model". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2009, pp. 935–942.

[200] Antoine Miech et al. "Howto100m: Learning a text-video embedding by watching hundred million narrated video clips". In: *Proc. ICCV*. 2019.

[201] Ben Mildenhall et al. "Nerf: Representing scenes as neural radiance fields for view synthesis". In: *Communications of the ACM* 65.1 (2021), pp. 99–106.

[202] Ravi Teja Mullapudi et al. "Online model distillation for efficient video inference". In: *Proc. CVPR*. 2019.

[203] Jonghwan Mun et al. "Marioqa: Answering questions by watching gameplay videos". In: *Proc. ICCV*. 2017.

[204] Daniel Neimark et al. "Video Transformer Network". In: *arXiv preprint arXiv:2102.00719* (2021).

[205] Alaaeldin El-Nouby et al. "Training Vision Transformers for Image Retrieval". In: *arXiv preprint arXiv:2102.05644* (2021).

[206] OpenAI. *Introducing ChatGPT*. 2022. URL: https://www.openai.com/research/chatgpt/.

[207] R OpenAI. "GPT-4 technical report". In: *arXiv* (2023).

[208] Tycho FA van der Ouderaa and Daniel E Worrall. "Reversible gans for memory-efficient image-to-image translation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4720–4728.

[209] Bo Pang et al. "Trajectory Prediction with Latent Belief Energy-Based Model". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 11814–11824.

[210] Kishore Papineni et al. “Bleu: a method for automatic evaluation of machine translation”. In: 2002.

[211] Mandela Patrick et al. “Keeping Your Eye on the Ball: Trajectory Attention in Video Transformers”. In: *NIPS* (2021).

[212] David A Patterson. “Latency lags bandwith”. In: *Communications of the ACM* 47.10 (2004), pp. 71–75.

[213] Stefano Pellegrini, Andreas Ess, and Luc Van Gool. “Improving data association by joint modeling of pedestrian trajectories and groupings”. In: *European conference on computer vision*. Springer. 2010, pp. 452–465.

[214] Stefano Pellegrini et al. “You’ll never walk alone: Modeling social behavior for multi-target tracking”. In: *2009 IEEE 12th International Conference on Computer Vision*. IEEE. 2009, pp. 261–268.

[215] Mihir Pendse et al. “Memory Efficient 3D U-Net with Reversible Mobile Inverted Bottlenecks for Brain Tumor Segmentation”. In: *International MICCAI Brainlesion Workshop*. Springer. 2020, pp. 388–397.

[216] Bas Peters, Eldad Haber, and Keegan Lensink. “Fully reversible neural networks for large-scale surface and sub-surface characterization via remote sensing”. In: *arXiv preprint arXiv:2003.07474* (2020).

[217] Mahima Pushkarna, Andrew Zaldivar, and Oddur Kjartansson. “Data cards: Purposeful and transparent dataset documentation for responsible ai”. In: *2022 ACM Conference on Fairness, Accountability, and Transparency*. 2022.

[218] Patrick Putzky and Max Welling. “Invert to learn to invert”. In: *Advances in Neural Information Processing Systems* 32 (2019), pp. 446–456.

[219] Zhaofan Qiu, Ting Yao, and Tao Mei. “Learning spatio-temporal representation with pseudo-3d residual networks”. In: *Proc. ICCV*. 2017.

[220] Alec Radford et al. “Learning transferable visual models from natural language supervision”. In: *arXiv preprint arXiv:2103.00020* (2021).

[221] Ilija Radosavovic et al. “Designing network design spaces”. In: *Proc. CVPR*. June 2020.

[222] Prajit Ramachandran et al. “Stand-alone self-attention in vision models”. In: *arXiv preprint arXiv:1906.05909* (2019).

[223] Aditya Ramesh et al. “Zero-Shot Text-to-Image Generation”. In: *arXiv preprint arXiv:2102.12092* (2021).

[224] Eike Rehder and Horst Kloeden. “Goal-directed pedestrian prediction”. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2015, pp. 50–58.

[225] Eike Rehder et al. “Pedestrian prediction by planning using deep neural networks”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 1–5.

[226] Shaoqing Ren et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *NIPS*. 2015.

[227] Nicholas Rhinehart et al. “PRECOG: PREdiction Conditioned On Goals in Visual Multi-Agent Settings”. In: *arXiv preprint arXiv:1905.01296* (2019).

[228] A Robicquet et al. “Learning Social Etiquette: Human Trajectory Prediction In Crowded Scenes”. In: *European Conference on Computer Vision (ECCV)*.

[229] Alexandre Robicquet et al. “Learning social etiquette: Human trajectory understanding in crowded scenes”. In: *European conference on computer vision*. Springer. 2016, pp. 549–565.

[230] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab et al. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4.

[231] Azriel Rosenfeld and Mark Thurston. “Edge and curve detection for visual scene analysis”. In: *IEEE Transactions on computers* 100.5 (1971), pp. 562–569.

[232] Andrey Rudenko et al. “Human Motion Trajectory Prediction: A Survey”. In: *arXiv e-prints*, arXiv:1905.06113 (2019). arXiv: 1905.06113.

[233] Amir Sadeghian et al. “Sophie: An attentive gan for predicting paths compliant to social and physical constraints”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1349–1358.

[234] Amir Sadeghian et al. “TrajNet: Towards a Benchmark for Human Trajectory Prediction”. In: *arXiv preprint* (2018).

[235] Arka Sadhu, Kan Chen, and Ram Nevatia. “Video question answering with phrases via semantic roles”. In: *arXiv preprint arXiv:2104.03762* (2021).

[236] Tim Salzmann et al. “Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data”. In: *arXiv preprint arXiv:2001.03093* (2020).

[237] Michael E Sander et al. “Momentum residual neural networks”. In: *arXiv preprint arXiv:2102.07870* (2021).

[238] Rajat Sen, Hsiang-Fu Yu, and Inderjit S Dhillon. “Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 4837–4846.

[239] Nasim Shafiee, Taskin Padir, and Ehsan Elhamifar. “Introvert: Human Trajectory Prediction via Conditional 3D Attention”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 16815–16825.

[240] Mike Zheng Shou et al. “Generic event boundary detection: A benchmark for event segmentation”. In: *Proc. CVPR*. 2021.

[241] Gunnar A Sigurdsson et al. "Hollywood in homes: Crowdsourcing data collection for activity understanding". In: *ECCV*. 2016.

[242] Karen Simonyan and Andrew Zisserman. "Two-stream convolutional networks for action recognition in videos". In: *NIPS*. 2014.

[243] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *Proc. ICLR*. 2015.

[244] Jascha Sohl-Dickstein et al. "Deep unsupervised learning using nonequilibrium thermodynamics". In: *International conference on machine learning*. PMLR. 2015, pp. 2256–2265.

[245] Mattia Soldan et al. "Mad: A scalable dataset for language grounding in videos from movie audio descriptions". In: *Proc. CVPR*. 2022.

[246] Xiaomeng Song et al. "Explore multi-step reasoning in video question answering". In: 2018.

[247] Yang Song, Chenlin Meng, and Stefano Ermon. "Mintnet: Building invertible neural networks with masked convolutions". In: *arXiv preprint arXiv:1907.07945* (2019).

[248] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. "UCF101: A dataset of 101 human actions classes from videos in the wild". In: *arXiv preprint arXiv:1212.0402* (2012).

[249] Bingfeng Sun and Jian Zhang. "Invertible Image Compressive Sensing". In: *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*. Springer. 2021, pp. 548–560.

[250] Chen Sun et al. "Actor-Centric Relation Network". In: *ECCV*. 2018.

[251] Yifan Sun et al. "Summarizing CPU and GPU design trends with product data". In: *arXiv preprint arXiv:1911.11313* (2019).

[252] Christian Szegedy et al. "Going Deeper with Convolutions". In: *Proc. CVPR*. 2015.

[253] Christian Szegedy et al. "Going deeper with convolutions". In: *Proc. CVPR*. 2015.

[254] Christian Szegedy et al. "Rethinking the Inception Architecture for Computer Vision". In: *arXiv:1512.00567* (2015).

[255] Mingxing Tan and Quoc V Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: *arXiv preprint arXiv:1905.11946* (2019).

[256] Makarand Tapaswi et al. "Movieqa: Understanding stories in movies through question-answering". In: *Proc. CVPR*. 2016.

[257] Meng Keat Christopher Tay and Christian Laugier. "Modelling smooth paths using gaussian processes". In: *Field and Service Robotics*. Springer. 2008, pp. 381–390.

[258] Yi Tay et al. "Sparse sinkhorn attention". In: *Proc. ICML*. PMLR. 2020, pp. 9438–9447.

[259] S Thrun, W Burgard, and D Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents series), ser. Intelligent robotics and autonomous agents*. 2005.

[260] Sebastian Thrun. "Probabilistic robotics". In: *Communications of the ACM* 45.3 (2002), pp. 52–57.

[261] Michael Tomasello et al. "Understanding and sharing intentions: The origins of cultural cognition". In: *Behavioral and brain sciences* 28.5 (2005), pp. 675–691.

[262] Hugo Touvron et al. "DeiT: Data-efficient Image Transformers". In: *arXiv preprint arXiv:2012.12877* (2020).

[263] Hugo Touvron et al. "Going deeper with image transformers". In: *Proc. ICCV*. 2021.

[264] Hugo Touvron et al. "Training data-efficient image transformers & distillation through attention". In: *arXiv preprint arXiv:2012.12877* (2020).

[265] Du Tran et al. "A Closer Look at Spatiotemporal Convolutions for Action Recognition". In: *Proc. CVPR*. 2018.

[266] Du Tran et al. "Video Classification with Channel-Separated Convolutional Networks". In: *Proc. ICCV*. 2019.

[267] Shimon Ullman. "The interpretation of structure from motion". In: *Proceedings of the Royal Society of London. Series B. Biological Sciences* 203.1153 (1979), pp. 405–426.

[268] Jeya Maria Jose Valanarasu et al. "Medical Transformer: Gated Axial-Attention for Medical Image Segmentation". In: *arXiv preprint arXiv:2102.10662* (2021).

[269] Pavan Vasishta, Dominique Vaufreydaz, and Anne Spalanzani. "Natural vision based method for predicting pedestrian behaviour in urban environments". In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2017, pp. 1–6.

[270] Ashish Vaswani et al. "Attention is all you need". In: *arXiv preprint arXiv:1706.03762* (2017).

[271] Anirudh Vemula, Katharina Muelling, and Jean Oh. *Social Attention: Modeling Attention in Human Crowds*. 2018. arXiv: 1710.04689 `[cs.RO]`.

[272] Paul Vicol et al. "Moviegraphs: Towards understanding human-centric situations from videos". In: *Proc. CVPR*. 2018.

[273] Huiyu Wang et al. "MaX-DeepLab: End-to-End Panoptic Segmentation with Mask Transformers". In: *arXiv preprint arXiv:2012.00759* (2020).

[274] Sinong Wang et al. "Linformer: Self-attention with linear complexity". In: *arXiv preprint arXiv:2006.04768* (2020).

[275] Wenhai Wang et al. "Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions". In: *arXiv preprint arXiv:2102.12122* (2021).

[276] Xiaolong Wang and Abhinav Gupta. "Videos as Space-Time Region Graphs". In: *Proc. ECCV*. 2018.

[277] Xiaolong Wang et al. "Non-local Neural Networks". In: *Proc. CVPR*. 2018.

[278] Xiaolong Wang et al. "Non-local neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7794–7803.

[279] Yi Wang et al. "InternVideo: General Video Foundation Models via Generative and Discriminative Learning". In: *arXiv preprint arXiv:2212.03191* (2022).

[280] Yuqing Wang et al. "End-to-End Video Instance Segmentation with Transformers". In: *arXiv preprint arXiv:2011.14503* (2020).

[281] Peter Whittle. *Hypothesis testing in time series analysis*. Vol. 4. Almqvist & Wiksells boktr., 1951.

[282] Samuel Williams, Andrew Waterman, and David Patterson. "Roofline: an insightful visual performance model for multicore architectures". In: *Communications of the ACM* 52.4 (2009), pp. 65–76.

[283] Samuel Webb Williams. *Auto-tuning performance on multicore computers*. University of California, Berkeley, 2008.

[284] Bichen Wu et al. "Visual transformers: Token-based image representation and processing for computer vision". In: *arXiv preprint arXiv:2006.03677* (2020).

[285] Bo Wu et al. "Star: A benchmark for situated reasoning in real-world videos". In: 2021.

[286] Chao-Yuan Wu and Philipp Krahenbuhl. "Towards long-form video understanding". In: *Proc. CVPR*. 2021.

[287] Chao-Yuan Wu et al. "Long-Term Feature Banks for Detailed Video Understanding". In: *Proc. CVPR*. 2019.

[288] Tianran Wu et al. "Transferring domain-agnostic knowledge in video question answering". In: *arXiv preprint arXiv:2110.13395* (2021).

[289] Yuxin Wu et al. *Detectron2*. https://github.com/facebookresearch/detectron2. 2019.

[290] Junbin Xiao et al. "Next-qa: Next phase of question-answering to explaining temporal actions". In: *Proc. CVPR*. 2021.

[291] Shuang Xiao, Zhan Wang, and John Folkesson. "Unsupervised robot learning to predict person motion". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 691–696.

[292] Saining Xie et al. "Aggregated residual transformations for deep neural networks". In: *Proc. CVPR*. 2017.

[293] Saining Xie et al. "Rethinking spatiotemporal feature learning for video understanding". In: *arXiv:1712.04851* (2017).

[294] Yu Xiong et al. "A graph-based framework to bridge movies and synopses". In: *Proc. CVPR*. 2019.

[295] Dejing Xu et al. "Video question answering via gradually refined attention over appearance and motion". In: 2017, pp. 1645–1653.

[296] Li Xu, He Huang, and Jun Liu. "Sutd-trafficqa: A question answering benchmark and an efficient network for video reasoning over traffic events". In: *Proc. CVPR*. 2021.

[297] Takuma Yagi et al. *Future Person Localization in First-Person Videos*. 2018. arXiv: 1711.11217 [cs.CV].

[298] Kota Yamaguchi et al. "Who are you with and where are you going?" In: *CVPR 2011*. IEEE. 2011, pp. 1345–1352.

[299] Kashu Yamazaki, Vidhiwar Singh Rathour, and T Le. "Invertible Residual Network with Regularization for Effective Medical Image Segmentation". In: *arXiv preprint arXiv:2103.09042* (2021).

[300] Antoine Yang et al. "Just ask: Learning to answer questions from millions of narrated videos". In: *Proc. ICCV*. 2021.

[301] Antoine Yang et al. "Zero-shot video question answering via frozen bidirectional language models". In: *arXiv preprint arXiv:2206.08155* (2022).

[302] Jieming Yang et al. "Image compact-resolution and reconstruction using reversible network". In: *IET Image Processing* 14.16 (2020), pp. 4376–4384.

[303] Pinci Yang et al. "AVQA: A Dataset for Audio-Visual Question Answering on Videos". In: 2022.

[304] Sen Yang et al. "TransPose: Towards Explainable Human Pose Estimation by Transformer". In: *arXiv preprint arXiv:2012.14214* (2020).

[305] Qinghao Ye et al. "mplug-owl: Modularization empowers large language models with multimodality". In: *arXiv preprint arXiv:2304.14178* (2023).

[306] Yunan Ye et al. "Video question answering via attribute-augmented attention network learning". In: *Proc. SIGIR*. 2017.

[307] Kexin Yi et al. "Clevrer: Collision events for video representation and reasoning". In: *arXiv preprint arXiv:1910.01442* (2019).

[308] Jun Yu et al. "Multimodal transformer with multi-view visual representation for image captioning". In: *IEEE transactions on circuits and systems for video technology* 30.12 (2019), pp. 4467–4480.

[309] Zhou Yu et al. "Activitynet-qa: A dataset for understanding complex web videos via question answering". In: 2019.

[310] Li Yuan et al. "Tokens-to-token vit: Training vision transformers from scratch on imagenet". In: *arXiv preprint arXiv:2101.11986* (2021).

[311] Ye Yuan et al. "AgentFormer: Agent-Aware Transformers for Socio-Temporal Multi-Agent Forecasting". In: *arXiv preprint arXiv:2103.14023* (2021).

[312] Zhenxun Yuan et al. "Temporal-Channel Transformer for 3D Lidar-Based Video Object Detection in Autonomous Driving". In: *arXiv preprint arXiv:2011.13628* (2020).

[313] Boxiang Yun et al. “SpecTr: Spectral Transformer for Hyperspectral Pathology Image Segmentation”. In: *arXiv preprint arXiv:2103.03604* (2021).

[314] Heeseung Yun et al. “Pano-avqa: Grounded audio-visual question answering on 360deg videos”. In: *Proc. ICCV*. 2021.

[315] Sangdoo Yun et al. “Cutmix: Regularization strategy to train strong classifiers with localizable features”. In: *Proc. ICCV*. 2019.

[316] Amir Zadeh et al. “Social-iq: A question answering benchmark for artificial social intelligence”. In: *Proc. CVPR*. 2019.

[317] Hongyi Zhang et al. “Mixup: Beyond empirical risk minimization”. In: *Proc. ICLR*. 2018.

[318] Chen Zhao et al. “Reˆ 2TAL: Rewiring Pretrained Video Backbones for Reversible Temporal Action Localization”. In: *arXiv preprint arXiv:2211.14053* (2022).

[319] Hang Zhao et al. “TNT: Target-driveN Trajectory Prediction”. In: *arXiv preprint arXiv:2008.08294* (2020).

[320] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. “Exploring self-attention for image recognition”. In: *Proc. CVPR*. 2020, pp. 10076–10085.

[321] Hengshuang Zhao et al. “Point transformer”. In: *arXiv preprint arXiv:2012.09164* (2020).

[322] Wentian Zhao et al. “Video question answering on screencast tutorials”. In: *arXiv preprint arXiv:2008.00544* (2020).

[323] Yuekai Zhao, Shuchang Zhou, and Zhihua Zhang. “Multi-split Reversible Transformers Can Enhance Neural Machine Translation”. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. 2021, pp. 244–254.

[324] Zaixiang Zheng et al. “Duplex Sequence-to-Sequence Learning for Reversible Machine Translation”. In: *arXiv preprint arXiv:2105.03458* (2021).

[325] Zhun Zhong et al. “Random erasing data augmentation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 07. 2020, pp. 13001–13008.

[326] Bolei Zhou et al. “Temporal Relational Reasoning in Videos”. In: *ECCV*. 2018.

[327] Tyler Zhu. “Speeding Up Reversible Vision Transformers”. In: *http://bit.ly/3J6Q0Cb* (2022).

[328] Brian D Ziebart et al. “Planning-based prediction for pedestrians”. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2009, pp. 3931–3936.

[329] Haosheng Zou et al. “Understanding human behaviors in crowds by imitating the decision-making process”. In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

# Appendix A

# Chapter 4 Supplementary Material

## A.1 Appendix

In this appendix, §A.2 contains further *ablations* for Kinetics (§A.2) & ImageNet (§A.2), §A.4 contains an *analysis* on computational complexity of MHPA, and §A.3 qualitative *observations* in MViT and ViT models. §A.5 contains additional *implementation details* for: Kinetics (§A.5), AVA (§A.5), Charades (§A.5), SSv2 (§A.5), and ImageNet (§A.5).

## A.2 Additional Results

### Results: Kinetics-700 Classification

| model | pretrain | top-1 | top-5 | GFLOPs×views | Param |
|---|---|---|---|---|---|
| SlowFast 16×8 +NL [78] | K600 | 71.0 | 89.6 | 234×3×10 | 59.9 |
| **MViT**-B, 16×4 | - | 71.2 | 90.0 | 70.5×1×5 | 36.8 |
| **MViT**-B-24, 32×3 | - | **74.0** | **91.7** | 236×1×5 | 52.9 |

Table A.1: **Comparison with previous work on Kinetics-700**.

**Kinetics-700** [37] is the largest version of Kinetics with 522k training videos. Results are in Table A.1. We train MViT from-scratch, without any pre-training. **MViT**-B, 16×4 achieves 71.2% top-1 accuracy already outperforming the best previous SlowFast [78] model. We further train a deeper 24-layer model with longer sampling, **MViT**-B-24, 32×3, which achieves 74.0% top-1 accuracy.

### Ablations: Kinetics-400 Classification

**Inference cost.** In the spirit of [75] we aim to provide further ablations for the effect of using *fewer* testing clips for efficient video-level inference. In Fig. A.1 we analyze the trade-off for the full

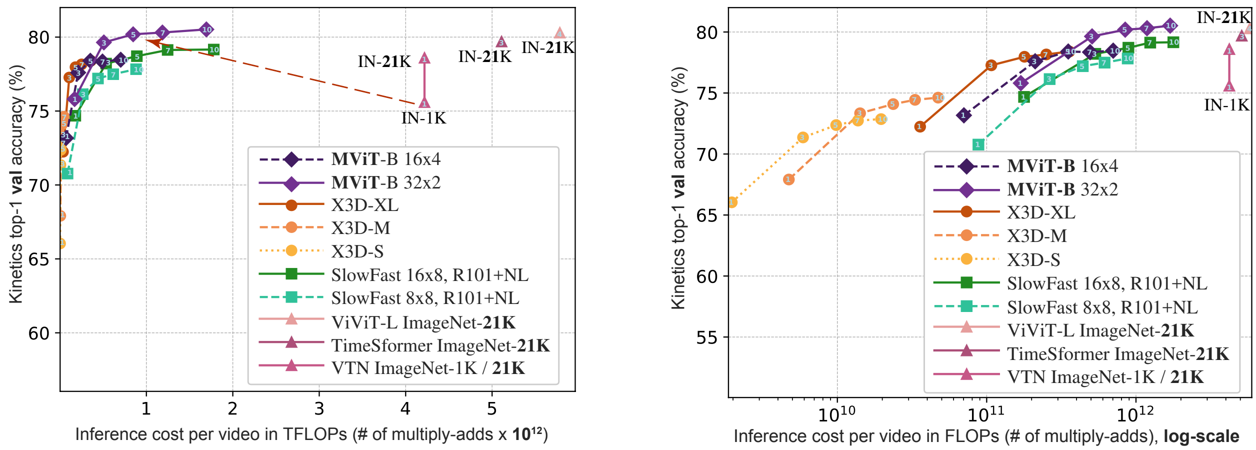



Figure A.1: **Accuracy/complexity trade-off** on K400-**val** for varying # of inference clips per video. The top-1 accuracy (vertical axis) is obtained by $K$-Center clip testing where the number of temporal clips $K \in \{1, 3, 5, 7, 10\}$ is shown in each curve. The horizontal axis measures the full inference cost per video. The left-sided plots show a linear and the right plots a logarithmic (**log**) scale.

inference of a video, when varying the number of temporal clips used. The vertical axis shows the top-1 accuracy on K400-val and the horizontal axis the overall inference cost in FLOPs for different model families: **MViT**, X3D [75], SlowFast [77], and concurrent ViT models, VTN [204] ViT-B-TimeSformer [22] ViT-L-ViViT [8], pre-trained on ImageNet-**21K**.

We first compare MViT with concurrent Transformer-based methods in the left plot in Fig. A.1. All these methods, VTN [204], TimeSformer [22] and ViViT [8], pre-train on ImageNet-**21K** and use the ViT [67] model with modifications on top of it. The inference FLOPs of these methods are around 5-10×higher than MViT models with equivalent performance; for example, ViT-L-ViViT [8] uses 4 clips of 1446G FLOPs (*i.e.* 5.78 TFLOPs) each to produce 80.3% accuracy while MViT-B, 32×3 uses 5 clips of 170G FLOPs (*i.e.* 0.85 TFLOPs) to produce 80.2% accuracy. Therefore, MViT-L can provide similar accuracy at 6.8× lower FLOPs (and 8.5× lower parameters), than concurrent ViViT-L [8]. More importantly, the MViT result is achieved *without external data*. All concurrent Transformer based works [204, 22, 8] require the huge scale ImageNet-21K to be competitive, and the performance degrades significantly (-3% accuracy, see IN-1K in Fig. A.1 for VTN [204]). These works further report failure of training without ImageNet initialization.

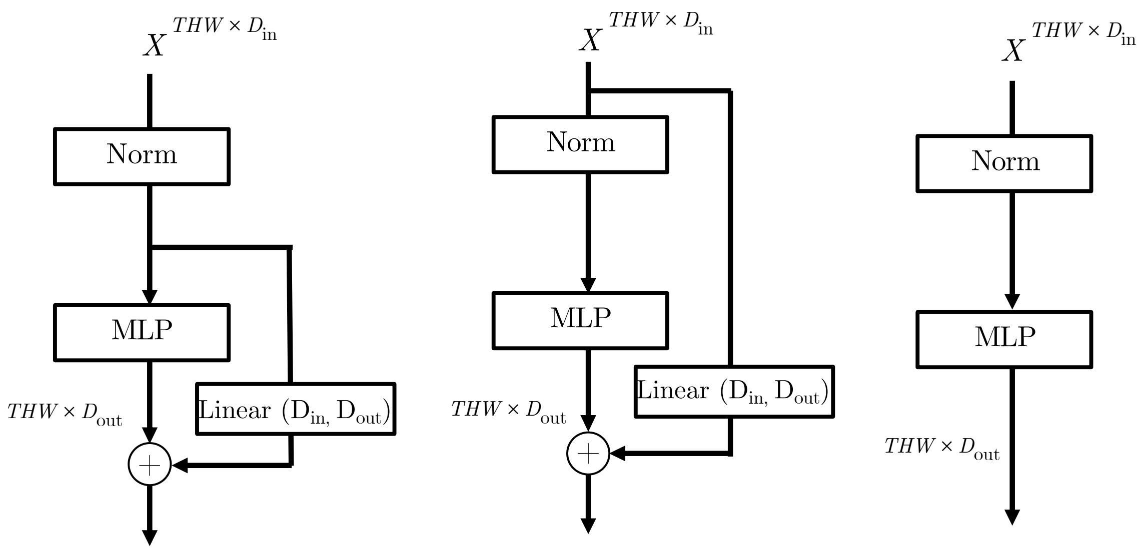
The plot in Fig. A.1 right shows this same plot with a logarithmic scale applied to the FLOPs axis. Using this scaling it is clearer to observe that smaller models convolutional models (X3D-S and X3D-M) can still provide more efficient inference in terms of multiply-add operations and MViT-B compute/accuracy trade-off is similar to X3D-XL.

**Ablations on skip-connections.** Recall that, at each scale-stage transition in MViT, we expand the channel dimension by increasing the output dimension of the previous stages' MLP layer; therefore, it is not possible to directly apply the original skip-connection design [67], because the input channel dimension ($D_{\text{in}}$) differs from the output channel dimension ($D_{\text{out}}$). We ablate three strategies for this:

(a) First normalize the input with layer normalization and then expand its channel dimension to match the output dimension with a linear layer (Fig. A.2a); this is our default.

(b) Directly expand the channel dimension of the input by using a linear layer to match the dimension (Fig. A.2b).

(c) No skip-connection for stage-transitions (Fig. A.2c).




(a) **normalized** skip-connection (b) unnormalized skip-connection (c) no skip-connection

Figure A.2: **Skip-connections at stage-transitions.** Three skip-connection variants for expanding channel dimensions: (a) first normalize the input with layer normalization (Norm) and then expand its channel dimension; (b) directly expand the channel dimension of the input; (c) no skip-connection at stage-transitions.

| | method | top-1 | top-5 |
|---|---|---|---|
| (a) | **normalized** skip-connection | **77.2** | **93.1** |
| (b) | unnormalized skip-connection | 74.6 | 91.3 |
| (c) | no skip-connection | 74.7 | 91.8 |

Table A.2: **Skip-connections at stage-transitions on K400.** We use our base model, MViT-B 16×4. Normalizing the skip-connection at channel expansion is essential for good performance.

Table A.2 shows the Kinetics-400 ablations for all 3 variants. Our default of using a normalized skip-connection (a) obtains the best results with 77.2% top-1 accuracy, while using an un-normalized skip-connection after channel expansion (b) decays significantly to 74.6% and using no skip-connection for all stage-transitions (c) has a similar result. We hypothesize that for expanding the channel dimension, normalizing the signal is essential to foster optimization, and use this design as our default in all other experiments.

| backbone | recipe | Acc |
|---|---|---|
| SlowFast R50, 8×8 | [77] | 77.0 |
| SlowFast R50, 8×8 | MViT | 67.4 |
| SlowFast R101, 8×8 | [77] | 78.0 |
| SlowFast R101, 8×8 | MViT | 61.6 |

Table A.3: **SlowFast models with MViT recipe on Kinetics-400.** The default recipe is using the recipe from the original paper. Accuracy is evaluated on 10×3 views.

**SlowFast with MViT recipe.** To investigate if our training recipe can benefit ConvNet models, we apply the same augmentations and training recipe as for MViT to SlowFast in Table A.3. The results suggest that SlowFast models do not benefit from the MViT recipe directly and more studies are required to understand the effect of applying our training-from-scratch recipe to ConvNets, as it seems higher capacity ConvNets (R101) perform worse when using our recipe.

## Ablations: ImageNet Image Classification

We carry out ablations on ImageNet with the **MViT**-B-16 model with 16 layers, and show top-1 accuracy (Acc) as well as computational complexity measured in GFLOPs (floating-point operations). We also report Parameters in M($10^6$) and training GPU memory in G($10^9$) for a batch size of 512.

| stride s | FLOPs | Mem | Acc |
|---|---|---|---|
| 8×8 | 7.2 | 9.0 | 81.6 |
| 4×4 | 7.8 | 11.9 | 82.5 |
| 2×2 | 9.0 | 13.2 | 81.8 |
| none | 10.4 | 17.3 | 82.3 |

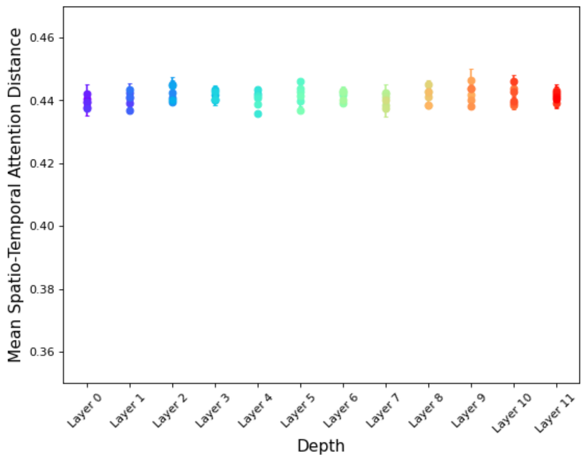
Table A.4: **ImageNet: Key-Value pooling**: We vary stride $s_H \times s_W$, for pooling $K$ and $V$. We use "adaptive" pooling that reduces stride w.r.t. stage resolution.

**Key-Value pooling for image classification.** The ablation in Table A.4 analyzes the pooling stride $\mathbf{s} = s_H \times s_W$, for pooling $K$ and $V$ tensors. Here, we use our default 'adaptive' pooling that uses a stride w.r.t. stage resolution, and keeps the $K, V$ resolution *fixed* across all stages.

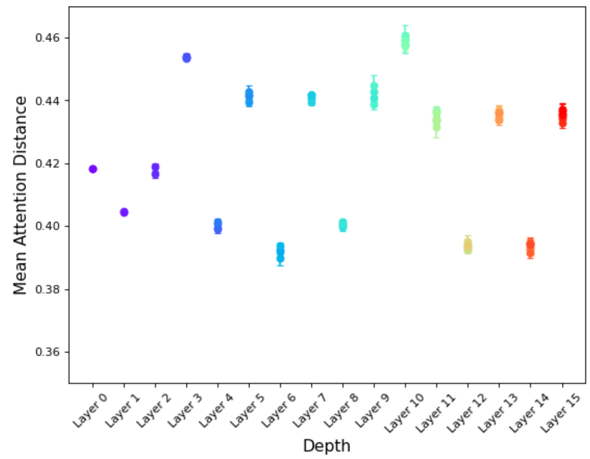
First, we compare the baseline which uses pooling with a fixed stride of 4×4 with a model has a stride of 8×8: this drops accuracy from 82.5% to 81.6%, and reduces FLOPs and memory by 0.6G and 2.9G.

Second, we reduce the stride to 2×2, which increases FLOPs and memory significantly but performs 0.7% *worse* than our default stride of 4×4.

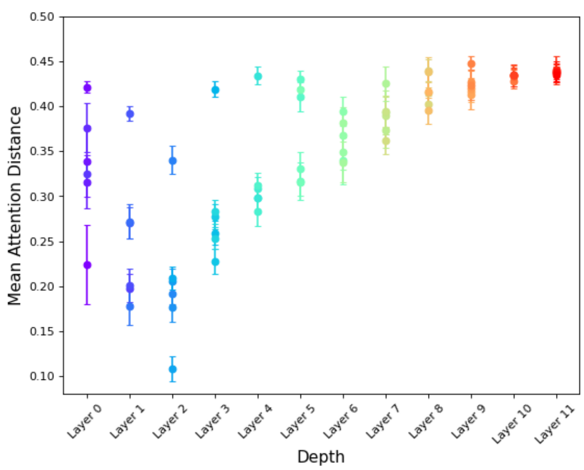
Third, we remove the $K, V$ pooling completely which increases FLOPs by 33% and memory consumption by 45%, while providing lower accuracy than our default.



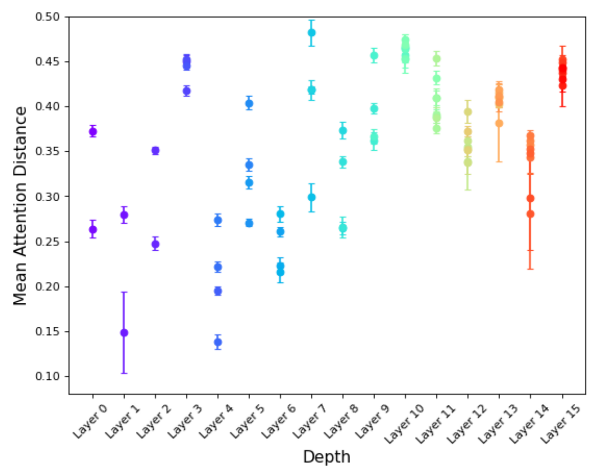

(a) ViT-B at **initialization**




(b) MViT-B at **initialization**




(c) ViT-B at **convergence**




(d) MViT-B at **convergence**

Figure A.3: **Mean attention distance** across layers ***at initialization/convergence*** for Vision Transformer (a)/(c) & Multiscale Vision Transformers (b)/(d). Each point shows the normalized average attention distance (weighted by the attention scores, with 1.0 being maximum possible distance) for each head in a layer. MViT attends close and distant features throughout the network hierarchy.

Overall, the results show that our $K, V$ pooling is an effective technique to *increase* accuracy *and decrease* cost (FLOPs/memory) for image classification.

## A.3 Qualitative Experiments: Kinetics

In Figure A.3, we plot the mean attention distance for all heads across all the layers of our Multiscale Transformer model and its Vision Transformer counterpart, at initialization with random weights, and at convergence after training. Each head represents a point in the plots (ViT-B has more heads). Both the models use the exact same weight initialization scheme and the difference in the attention signature stems purely from the multiscale skeleton in MViT. We observe that the dynamic range of attention distance is about **4**× larger in the MViT model than ViT *at initialization* itself (A.3a *vs*. A.3b). This signals the strong inductive bias stemming from the multiscale design of MViT. Also note that while at initialization, every layer in ViT has roughly the same mean attention distance, the MViT layers have strikingly different mean attention signatures indicating distinct predilections towards global and local features.

The bottom row of Fig. A.3 shows the same plot for a converged Vision Transformer (A.3c) and Multiscale Vision Transformer (A.3d) model.

We notice very different trends between the two models *after training*. While the ViT model (A.3c) has a consistent increase in attention distance across layers, the MViT model (A.3d) is not monotonic at all. Further, the intra-head variation in the ViT model decreases as the depth saturates, while, for MViT, different heads are still focusing on different features even in the higher layers. This suggests that some of the capacity in the ViT model might indeed be wasted with redundant computation while the lean MViT heads are more judiciously utilizing their compute. Noticeable is further a larger delta (between initialization in Fig. A.3a and convergence in A.3c) in the overall attention distance signature in the ViT model, compared to MViT's location distribution.

## A.4 Computational Analysis

Since attention is quadratic in compute and memory complexity, pooling the key, query and value vectors have direct benefits on the fundamental compute and memory requirements of the pooling operator and by extension, on the complete Multiscale Transformer model. Consider an input tensor of dimensions $T \times H \times W$ and corresponding sequence length $L = T \cdot H \cdot W$. Further, assume the key, query and value strides to be $\mathbf{s}^K$, $\mathbf{s}^Q$ and $\mathbf{s}^V$. As described in Sec. 3.1 in main paper, each of the vectors would experience a spatiotemporal resolution downsampling by a factor of their corresponding strides. Equivalently, the sequence length of query, key and value vectors would be reduced by a factor of $f^Q$, $f^K$ and $f^V$ respectively, where,

$$f^j = s^j_T \cdot s^j_H \cdot s^j_W, \; \forall \, j \in \{Q, K, V\}.$$

**Computational complexity.** Using these shorter sequences yields a corresponding reduction in space and runtime complexities for the pooling attention operator. Considering key, query and value vectors to have sequence lengths $L/f_k$, $L/f_q$ and $L/f_v$ after pooling, the overall runtime complexity of computing the key, query and value embeddings is $O(THWD^2/h)$ per head, where $h$ is the number of heads in MHPA. Further, the runtime complexity for calculating the full attention matrix and the weighed sum of value vectors with reduced sequence lengths is $O(T^2H^2W^2D/f_qf_hh)$ per head. Computational complexity for pooling is

$$T(\mathcal{P}(\cdot;\boldsymbol{\Theta})) = O\left(THW \cdot D \cdot \frac{k_T k_W k_H}{s_T s_W s_H}\right),$$

which is negligible compared to the quadratic complexity of the attention computation and hence can be ignored in asymptotic notation. Thus, the final runtime complexity of MHPA is $O(THWD(D + THW/f_qf_k))$.

**Memory complexity.** The space complexity for storing the sequence itself and other tensors of similar sizes is $O(THWD)$. Complexity for storing the full attention matrix is $O(T^2H^2W^2h/f_qf_k)$. Thus the total space complexity of MHPA is $O(THWh(D/h + THW/f_qf_k))$.

**Design choice.** Note the trade-off between the number of channels $D$ and the sequence length term $THW/f_qf_k$ in both space and runtime complexity. This tradeoff in multi head pooling attention informs two critical design choices of Multiscale Transformer architecture.

First, as the effective spatiotemporal resolution decreases with layers because of diminishing $THW/f_qf_k$, the channel capacity is increased to keep the computational time spent (FLOPs) roughly the same for each stage.

Second, for a fixed channel dimension, $D$, higher number of heads $h$ cause a prohibitively larger memory requirement because of the $(D + h * THW/f_qf_k)$ term. Hence, Multiscale Transformer starts with a small number of heads which is increased as the resolution factor $THW/f_qf_k$ decreases, to hold the effect of $(D + h * THW/f_qf_k)$ roughly constant.

## A.5 Additional Implementation Details

We implement our model with PySlowFast [72]. Code and models are available at: https://github.com/facebookresearch/SlowFast.

### Details: Kinetics Action Classification

**Architecture details.** As in original ViT [67], we use residual connections [111] and Layer Normalization (LN) [11] in the pre-normalization configuration that applies LN at the beginning of the residual function, and our MLPs consist of two linear layers with GELU activation [119], where the first layer expands the dimension from $D$ to $4D$, and the second restores the input dimension $D$, except at the end of a scale-stage, where we increase this channel dimensions to match the input of the next scale-stage. At such stage-transitions, our skip connections receive an extra linear layer that takes as input the layer-normalized signal which is also fed into the MLP. In case of $Q$-pooling at scale-stage transitions, we correspondingly pool the skip-connection signal.

**Optimization details.** We use the truncated normal distribution initialization in [109] and adopt synchronized AdamW [189] training on 128 GPUs following the recipe in [264, 77]. For Kinetics, we train for 200 epochs with 2 repeated augmentation [124] repetitions. The mini-batch size is 4 clips per GPU (so the overall $\text{batchsize}$ is $512$).

We adopt a half-period cosine schedule [190] of learning rate decaying: the learning rate at the $n$-th iteration is $\eta \cdot 0.5[\cos(\frac{n}{n_{\text{max}}}\pi) + 1]$, where $n_{\text{max}}$ is the maximum training iterations and the base learning rate $\eta$ is set as $1.6 \cdot 10^{-3}$. We linearly scale the base learning rate w.r.t. the overall batch-size, $\eta = 1.6 \cdot 10^{-3} \frac{\text{batchsize}}{512}$, and use a linear warm-up strategy in the first 30 epochs [96]. The cosine schedule is completed when reaching a final learning rate of $1.6 \cdot 10^{-5}$. We extract the class token after the last stage and use it as the input to the final linear layer to predict the output classes. For **Kinetics-600** all hyper-parameters are identical to K400.

**Regularization details.** We use weight decay of $5 \cdot 10^{-2}$, a dropout [120] of 0.5 before the final classifier, label-smoothing [254] of 0.1 and use stochastic depth [130] (*i.e*. drop-connect) with rate 0.2.

Our data augmentation is performed on input clips by applying the same transformation across all frames. To each clip, we apply a random horizontal flip, Mixup [317] with $\alpha = 0.8$ to half of the clips in a batch and CutMix [315] to the other half, Random Erasing [325] with probability $0.25$, and Rand Augment [54] with probability of $0.5$ for $4$ layers of maximum magnitude $7$.

For the temporal domain, we randomly sample a clip from the full-length video, and the input to the network are $T$ frames with a temporal stride of $\tau$; denoted as $T \times \tau$ [77]. For the spatial domain, we use Inception-style [252] cropping that randomly resizes the input *area* between a [min, max], scale of [0.08, 1.00], and jitters aspect ratio between 3/4 to 4/3, before taking an $H \times W$ = 224×224 crop.

**Fine-tuning from ImageNet.** To fine-tune our ViT-B baseline, we extend it to take a video clip of $T = 8$ frames as input and initialize the model weights from the ViT-B model [67] pre-trained on ImageNet-21K dataset. The positional embedding is duplicated for each frame. We fine-tune the model for 30 epochs with SGD using the recipe in [77]. The mini-batch size is 2 clips per GPU and a half-period cosine learning rate decay is used. We linearly scale the base learning rate w.r.t. the overall batch-size, $\eta = 10^{-3} \frac{\text{batchsize}}{16}$. Weight decay is set to $10^{-4}$.

## Details: AVA Action Detection

**Dataset.** The AVA dataset [103] has bounding box annotations for spatiotemporal localization of (possibly multiple) human actions. It has 211k training and 57k validation video segments. We follow the standard protocol reporting mean Average Precision (mAP) on 60 classes [103] on AVA v2.2.

**Detection architecture.** We follow the detection architecture in [77] to allow direct comparison of MViT against SlowFast networks as a backbone.

First, we reinterpret our transformer spacetime cube outputs from MViT as a spatial-temporal feature map by concatenating them according to the corresponding temporal and spatial location.

Second, we employ a the detector similar to Faster R-CNN [226] with minimal modifications adapted for video. Region-of-interest (RoI) features [89] are extracted at the generated feature map from MViT by extending a 2D proposal at a frame into a 3D RoI by replicating it along the temporal axis, similar as done in previous work [103, 250, 140], followed by application of frame-wise RoIAlign [114] and temporal global average pooling. The RoI features are then max-pooled and fed to a per-class, sigmoid classifier for prediction.

**Training.** We initialize the network weights from the Kinetics models and adopt synchronized SGD training on 64 GPUs. We use 8 clips per GPU as the mini-batch size and a half-period cosine schedule of learning rate decaying. The base learning rate is set as $0.6$. We train for 30 epochs with linear warm-up [96] for the first 5 epochs and use a weight decay of $10^{-8}$ and stochastic depth [130] with rate 0.4. Ground-truth boxes, and proposals overlapping with ground-truth boxes by IoU $> 0.9$, are used as the samples for training. The region proposals are identical to the ones used in [77].

**Inference.** We perform inference on a single clip with $T$ frames sampled with stride $\tau$ centered at the frame that is to be evaluated.

## Details: Charades Action Classification

**Dataset.** Charades [241] has ~9.8k training videos and 1.8k validation videos in 157 classes in a multi-label classification setting of longer activities spanning ~30 seconds on average. Performance is measured in mean Average Precision (mAP).

**Training.** We fine-tune our MViT models from the Kinetics models. A per-class sigmoid output is used to account for the multi-class nature. We train with SGD on 32 GPUs for 200 epochs using 8 clips per GPU. The base learning rate is set as 0.6 with half-period cosine decay. We use weight decay of $10^{-7}$ and stochastic depth [130] with rate 0.45. We perform the same data augmentation schemes as for Kinetics in §A.5, except of using Mixup.

**Inference.** To infer the actions over a single video, we spatiotemporally max-pool prediction scores from multiple clips in testing [77].

## Details: Something-Something V2 (SSv2)

**Dataset.** The Something-Something V2 dataset [97] contains 169k training, and 25k validation videos. The videos show human-object interactions to be classified into 174 classes. We report accuracy on the validation set.

**Training.** We fine-tune the pre-trained Kinetics models. We train for 100 epochs using 64 GPUs with 8 clips per GPU and a base learning rate of 0.02 with half-period cosine decay [190]. Weight decay is set to $10^{-4}$ and stochastic depth rate [130] is 0.4. Our training augmentation is the same as in §A.5, but as SSv2 requires distinguishing between directions, we disable random flipping in training. We use segment-based input frame sampling [177] that splits each video into segments, and from each of them, we sample one frame to form a clip.

**Inference.** We take single clip with 3 spatial crops to form predictions over a single video in testing.

## Details: ImageNet

**Datasets.** For image classification experiments, we perform our experiments on ImageNet-1K [57] dataset that has ~1.28M images in 1000 classes. We train models on the train set and report top-1 and top-5 classification accuracy (%) on the val set. Inference cost (in FLOPs) is measured from a single center-crop with resolution of $224^2$ if the input resolution was not specifically mentioned.

**Training.** We use the training recipe of DeiT [264] and summarize it here for completeness. We train for $100$ epochs with $3$ repeated augmentation [124] repetitions (overall computation equals $300$ epochs), using a batch size of $4096$ in $64$ GPUs. We use truncated normal distribution initialization [109] and adopt synchronized AdamW [189] optimization with a base learning rate of $0.0005$ per $512$ batch-size that is warmed up and decayed as half-period cosine, as in [264]. We use a weight decay of $0.05$, label-smoothing [254] of $0.1$. Stochastic depth [130] (*i.e.* drop-connect) is also used with rate $0.1$ for model with depth of 16 (MViT-B-16), and rate $0.3$ for deeper models (MViT-B-24). Mixup [317] with $\alpha = 0.8$ to half of the clips in a batch and CutMix [315] to the other half, Random Erasing [325] with probability $0.25$, and Rand Augment [54] with maximum magnitude $9$ and probability of $0.5$ for $4$ layers (for max-pooling) or $6$ layers (for conv-pooling).

# Appendix B

# Chapter 5 Supplementary Material

## B.1 Architecture Details

**Reversible Vision Transformers** Table A.1 shows the architectures for all the Reversible Vision Transformer Models. All models closely follow the original ViT architectures [68] in matched performance, parameters, FLOPs and much lower memory footprint (Table 5.1). Output sizes denote the tensor shapes of the two residual streams at the end of each reversible Vision Transformer block. Note that even though the intermediate activations are twice the non-reversible variant, the actual memory needed is much lower because of memory reuse in reversible training. Further, the FLOPs are matched since each layer is performed only one of the two streams.

**Reversible Multiscale Vision Transformers** Table A.2 shows the architecture for the Rev-MViT-B model for image classification. The backbone is made-up of two stages – Stage-transition blocks that increase the channel capacity and down-sample the resolution and the reversible Stage-preserving blocks that perform the majority of computation without changing feature dimensions. Similar to Rev-ViT, the output sizes of both the streams are denoted. Fusion blocks operate on $Y_1$ and $Y_2$ together, hence operate with computationally light operations (Table 5.6).

## B.2 Training Settings

**ImageNet.** Table A.3 shows the training recipes for ViT-L and Rev-ViT-L models presented in Table 5.1. Note that ViT-L is quite heavy with 61.6 GFLOPs and hence we adopt a shorter 200 epochs recipe for faster experiment cycle for developing Rev-ViT-L. Smaller ViT models – ViT-S and ViT-B – are trained according to the Data efficient transformers [262] and are all trained for 300 epochs. Hence, the accuracy difference between ViT-L which achieves 81.5% while ViT-B achieves 81.8% overall. MViT-B model follows the 300 epochs recipe as well proposed in [71].

**Kinetics-400 & Kinetics-600.** We follow the recipes proposed in [71] to train the Rev-MViT-B architecture (Table A.2) following crucial modifications shown in Table 5.5.

| stage | operators | output sizes |
|---|---|---|
| data | | 82242241 |
| patch | 1×16×16, 384<br>stride 1×16×16 | 384814142 |
| rev | $\begin{bmatrix} \text{MHA}(384) \\ \text{MLP}(1536) \end{bmatrix} \times 12$ | 3841414 |

(a) **Rev-ViT**-S with **4.6**G FLOPs, **22**M param, **8.8**MB/img memory, and **79.9**% top-1 accuracy.

| stage | operators | output sizes |
|---|---|---|
| data | | 82242241 |
| patch | 1×16×16, 768<br>stride 1×16×16 | 768814142 |
| rev | $\begin{bmatrix} \text{MHA}(768) \\ \text{MLP}(3072) \end{bmatrix} \times 12$ | 7681414 |

(b) **Rev-ViT**-B with **17.6**G FLOPs, **87**M param, **17**MB/img memory, and **81.8**% top-1 accuracy.

| stage | operators | output sizes |
|---|---|---|
| data | | 82242241 |
| patch | 1×16×16, 1024<br>stride 1×16×16 | 1024814142 |
| rev | $\begin{bmatrix} \text{MHA}(1024) \\ \text{MLP}(4096) \end{bmatrix} \times 24$ | 10241414 |

(c) **Rev-ViT**-L with **61.6**G FLOPs, **305**M param, **22.6**MB/img memory, and **81.4**% top-1 accuracy.

Table A.1: **Reversible Vision Transformer Architectures**: Rev-ViT are reversible adaption of ViT with exactly matched FLOPs, parameters and accuracy under identical conditions but with much lower GPU memory footprints.

**MS-COCO.** For object detection experiments, we adopt the Mask R-CNN [114] object detection framework in Detectron2 [289]. We follow the same training settings from [186], AdamW optimizer [189] ($\beta_1, \beta_2 = 0.9, 0.999$, base learning rate $1.6{-}4$ for base size of 64, and weight decay of 0.1), and 3x schedule (36 epochs). The drop path rate is set as $0.4$. We use PyTorch's automatic mixed precision during training.

# Acknowledgements

The authors would like to thank Harshayu Girase for help with benchmarking models, Amir Gholami, Ajay Jain and Nikita Kiatev for helpful research discussions and reference suggestions, Ajay Jain, Matthew Tancik and Hang Gao for writing discussions and Shubh Gupta, Suzie Petryk, Hang Gao, Abhinav Agarwal, Medhini Narasimhan and Amur Ghosh for proofreading the manuscript.

| stage | operators | output sizes |
|---|---|---|
| data | | 162242241 |
| cubification | 7×7, 96<br>stride 4×4 | 96856562 |
| Stage-Preserving | [MHPA(96)<br>MLP(384)] ×1 | 965656 |
| Stage-Transition | 1927681 | 192828283 |
| Stage-Preserving | [MHPA(192)<br>MLP(768)] ×1 | 1922828 |
| Stage-Transition | 38415361 | 384814143 |
| Stage-Preserving | [MHPA(384)<br>MLP(1536)] ×10 | 3841414 |
| Stage-Transition | 76830721 | 7688773 |
| Stage-Preserving | [MHPA(768)<br>MLP(3072)] ×1 | 76877 |

Table A.2: **Rev-MViT**-B with **8.7**G FLOPs, **39**M param, **66.8**MB/img memory, and **82.5**% top-1 accuracy is reversible adaption of MViT-B architecture [68].

| Training Hyperparameter | ViT-B | Rev-ViT-B |
|---|---|---|
| Learning Rate | 1e-4 | 7e-5 |
| Random augment Repeats (N) | 1 | 2 |
| Random augment Magnitude (M) | 9 | 7 |
| Optimizer Momentum | (0.9, 0.95) | (0.9, 0.999) |
| Weight Decay | 0.3 | 0.3 |
| Batch Size | 4096 | 4096 |
| Epochs | 200 | 200 |
| Label Smoothing | 0.1 | 0.1 |
| Drop Path Rate | 0.2 | 0.2 |
| Mixup | 0.8 | 0.8 |
| Cutmix | 1.0 | 1.0 |

Table A.3: **Training Recipe for ViT-L and Rev-ViT-L**

# Appendix C

# Chapter 6 Supplementary Material

## EgoSchema Datasheet

**Motivation**

**For what purpose was the dataset created?** Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description.

EgoSchema is a diagnostic benchmark for assessing very long-form video-language understanding capabilities of modern multimodal systems. While some prior works have proposed video datasets with long clip lengths, we posit that merely the length of the video clip does not truly capture the temporal difficulty of the video task that is being considered. To remedy this, we introduce temporal certificate sets, a general notion for capturing the intrinsic temporal understanding length associated with a broad range of video understanding tasks & datasets. Please see Section 6.3 in the main paper for more details.

**Who created this dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?**

The authors created the dataset within the Malik Group at Berkeley AI Research, UC Berkeley. The authors created it for the public at large without reference to any particular organization or institution.

## Composition

**What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)?** Are there multiple types of instances (e.g., movies, users, and ratings; people and interactions between them; nodes and edges)? Please provide a description.

Each instance in the dataset represents a 3-minute video and text that contains a question and five answer options.

**How many instances are there in total (of each type, if appropriate)?**

EgoSchema has a total of 5063 instances each containing one video, one question, and five answer options. You can see further statistics on the whole data on our website egoschema.github.io.

**Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set?** If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g., geographic coverage)? If so, please describe how this representativeness was validated/verified. If it is not representative of the larger set, please describe why not (e.g., to cover a more diverse range of instances, because instances were withheld or unavailable).

The video component of our dataset derives from the broader Ego4D dataset. For our research, we selectively extracted non-overlapping three-minute segments from the Ego4D video data, each segment consisting of a minimum of 30 human-annotated narrations (where each narration refers to a timestamped sentence). Detailed statistic of the number of viable clips for different possible length and narration density choices is discussed in Supplementary Section C.1. The selected subset is very diverse in human behavior as can be seen by the activity statistics presented on egoschema.github.io.

**What data does each instance consist of? "Raw" data (e.g., unprocessed text or images) or features?** In either case, please provide a description.

Each instance in our dataset comprises raw mp4 video data, captured at a rate of 30 frames per second and with a high resolution. Accompanying this video data, there are six text elements - one question and five corresponding answer options one of which is marked as the correct answer to the question.

**Is there a label or target associated with each instance?** If so, please provide a description.

Each instance is associated with a label ranging from 1 to 5 that indicates which of the five answer options is correct.

**Is any information missing from individual instances?** If so, please provide a description, explaining why this information is missing (e.g. because it was unavailable). This does not include intentionally removed information but might include, e.g., redacted text.

All instances are complete.

**Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)?** If so, please describe how these relationships are made explicit.

Some instances may have the same video but different questions and answers. It will be indicated by a clip unique identifier in the final dataset.

**Are there recommended data splits (e.g., training, development/validation, testing)?** If so, please provide a description of these splits, explaining the rationale behind them.

EgoSchema is designed specifically for zero-shot testing. Its primary purpose is to be able to asses the out of the box long-term video-language understanding capabilities of modern multimodal models.

**Are there any errors, sources of noise, or redundancies in the dataset?** If so, please provide a description.

The dataset was very carefully manually curated to mitigate any incidence of errors within the questions and answers. Although different questions may be posed for the same clip, it is ensured that there is no overlap between any two distinct clips. Further related details are also discussed in the limitations section in the main paper.

**Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)?** If it links to or relies on external resources, a) are there guarantees that they will exist, and remain constant, over time; b) are there official archival versions of the complete dataset (i.e., including the external resources as they existed at the time the dataset was created); c) are there any restrictions (e.g., licenses, fees) associated with any of the external resources that might apply to a future user? Please provide descriptions of all external resources and any restrictions associated with them, as well as links or other access points, as appropriate.

Entirety of the dataset will be made publicly available at our project website egoschema.github.io. We will also provide a download tool for preprocessing all the videos such as cutting clips, associating the question/answer text etc. Text will be released in a JSON format, hosted on our github repository. EgoSchema will be publicly released under the Ego4D license, which allows public use of the video and text data for both research and commercial purposes.

**Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor-patient confidentiality, data that includes the content of individuals non-public communications)?** If so, please provide a description.

No

**Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?** If so, please describe why.

No

**Does the dataset relate to people?** If not, you may skip the remaining questions in this section.

Some videos do contain people. However, the Ego4D authors employed an array of de-identification procedures primarily centered on ensuring a controlled environment with informed consent from all participants, and, where applicable, in public spaces with faces and other personally identifiable information suitably obscured. We strictly import all RGB information from Ego4D without any addition of our own.

**Does the dataset identify any subpopulations (e.g., by age, gender)?** If so, please describe how these subpopulations are identified and provide a description of their respective distributions within the dataset.
No

**Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset?** If so, please describe how.
No, Ego4D has employed an array of deidentification procedures in order to obscure any personally identifiable information such as people's faces.

**Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals racial or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)?** If so, please provide a description.
No

## Collection Process

**How was the data associated with each instance acquired?** Was the data directly observable (e.g., raw text, movie ratings), reported by subjects (e.g., survey responses), or indirectly inferred/derived from other data (e.g., part-of-speech tags, model-based guesses for age or language)? If data was reported by subjects or indirectly inferred/derived from other data, was the data validated/verified? If so, please describe how.

The video data, which is directly observable, was procured from the publicly accessible Ego4D dataset. In contrast, the text data was generated through the use of Large Language Models (LLMs) including GPT4, BARD, and Claude. These LLMs employed visual narrations from each video within the Ego4D dataset to generate the corresponding text.

**What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)?** How were these mechanisms or procedures validated?

The video and narration data were downloaded in accordance with the official Ego4D guidelines for data access: https://ego4d-data.org/docs/start-here. For the generation of the text data within our dataset, we utilized API access for GPT4 via OpenAI, for BARD via Google, and for Claude via Anthropic. This allowed us to generate three distinct questions for each video clip sampled from the Ego4D dataset. Upon the generation of these questions for each sampled video clip, we implemented a series of filtering procedures including Rule-based filtering, Blind filtering, and Manual curation. See Section 3.1.2 in the main paper for a more detailed explanation.

**If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)?**

The video component of our dataset derives from the broader Ego4D dataset. For our research, we selectively extracted non-overlapping three-minute segments from the Ego4D video data, each segment consisting of a minimum of 30 human-annotated narrations (where each narration refers to a timestamped sentence). Detailed statistic of the number of viable clips for different possible length and narration density choices is discussed in Supplementary Section C.1.

**Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?**

Our research utilized the services of Quantigo, a specialized data labelling company. The teams of Quantigo employees that were based in Bangladesh were compensated at a rate of 5 dollars per hour, at a wage significantly higher than the market hourly rate in Bangladesh. This was done to ensure fair compensation for the complex tasks performed while also contributing to the highest quality of the work delivered. It's important to note that our collaboration with Quantigo followed ethical guidelines, with the fair treatment of all employees involved and the appropriate respect for their expertise and labor. For exact instructions for human curation, see Supplementary Section C.2.

**Over what timeframe was the data collected? Does this timeframe match the creation timeframe of the data associated with the instances (e.g., recent crawl of old news articles)?** If not, please describe the timeframe in which the data associated with the instances was created.

The original videos within the Ego4D dataset were collected across various occasions spanning from 2019 to 2021. As for the EgoSchema, the textual information was collected over several sprints during the first half of 2023 based on the Ego4D narrations.

**Were any ethical review processes conducted (e.g., by an institutional review board)?** If so, please provide a description of these review processes, including the outcomes, as well as a link or other access point to any supporting documentation.

No

**Does the dataset relate to people?** If not, you may skip the remaining questions in this section.

Yes

**Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources (e.g., websites)?**

The video and narration data were acquired in accordance with the official Ego4D guidelines for data access: https://ego4d-data.org/docs/start-here/. The Ego4D authors had in turn ensured consent of the people involved.

**Were the individuals in question notified about the data collection?** If so, please describe (or show with screenshots or other information) how notice was provided, and provide a link or other access point to, or otherwise reproduce, the exact language of the notification itself.

Ego4d paper followed several procedures to ensure the preservation of privacy and the upholding of ethical standards. Notably, these procedures included obtaining informed consent from those wearing the cameras and adhering to de-identification requirements for personally identifiable information (PII). Given that the video collection was conducted by Ego4D, we are not in a position to provide specific instructions that were given to the camera wearers. The Ego4D privacy statement is available at https://ego4d-data.org/pdfs/Ego4D-Privacy-and-ethics-consortium-statement.pdf

**Did the individuals in question consent to the collection and use of their data?** If so, please describe (or show with screenshots or other information) how consent was requested and provided, and provide a link or other access point to, or otherwise reproduce, the exact language to which the individuals consented.

Ego4d paper privacy procedures have included obtaining informed consent from those wearing the cameras. Given that the video collection was conducted by Ego4D, we are not in a position to provide specific instructions that were given to the camera wearers. See Ego4D privacy statement.

**If consent was obtained, were the consenting individuals provided with a mechanism to revoke their consent in the future or for certain uses?** If so, please provide a description, as well as a link or other access point to the mechanism (if appropriate).

Ego4d paper privacy procedures have included allowing camera users to ask questions and withdraw at any time. Additionally, they were free to review and redact their own video. Given that the video collection was conducted by Ego4D, we are not in a position to provide specific instructions that were given to the camera wearers. You can find the Ego4D privacy statement at https://ego4d-data.org/pdfs/Ego4D-Privacy-and-ethics-consortium-statement.pdf.

**Has an analysis of the potential impact of the dataset and its use on data subjects (e.g., a data protection impact analysis) been conducted?** If so, please provide a description of this analysis, including the outcomes, as well as a link or other access point to any supporting documentation.

While we recognize the importance of this topic, we would, once more, refer to the Ego4D paper for an in-depth discussion. Ego4D acknowledges the potential privacy risks associated with the use of wearable devices in data collection and has taken several steps such as depersonalizing any sensitive information, blurring out faces and bodies, etc. towards maintaining privacy. The same carries over to the video data in EgoSchema as well. Broadly, very long-form video understanding is a core capability for agents that are to perceive the natural visual world. Hence, developing datasets such as EgoSchema will be critical to unlocking this key AI capability. Additionally, according to Ego4D privacy statement, all videos from Ego4D were reviewed by an approved member of one of the participant's universities or institutes to identify and assess potential privacy concerns.

## Preprocessing/cleaning/labeling

**Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)?** If so, please provide a description. If not, you may skip the remainder of the questions in this section.

The set of generated questions and answers from output was filtered by those LLMs and finally curated by humans. A detailed description can be found in Section 3. There was no preprocessing done on the video clips sampled from Ego4D.

**Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)?** If so, please provide a link or other access point to the "raw" data.

Human curation was employed to rectify errors in the question-answer sets, particularly cases where the identified correct answer was wrong or a wrong answer was actually correct. Given the crucial role of this step in ensuring the accuracy of our dataset, we do not find it necessary to release a version of the dataset prior to human curation. However, all the discarded "raw" data is indeed also saved.

**Is the software used to preprocess/clean/label the instances available?** If so, please provide a link or other access point.

The APIs for the Large Language Models (LLMs) are publicly accessible. The prompts for filtering and instructions for human curation are provided in Supplementary Section C and Supplementary Section C.2 respectively. Additionally all necessary code for generation, filtering etc. is provided in the supplementary materials.

## Distribution

**Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created?** If so, please provide a description.
The dataset will be made publicly available and can be used for both research and commercial purposes under the Ego4D license.

**How will the dataset be distributed (e.g., tarball on website, API, GitHub)** Does the dataset have a digital object identifier (DOI)?
The dataset will be distributed as a JSON file describing the unique identifier for each clip, the associated question, the five answer options, the label, and additional clip information that facilitates the tracing of the clip back to the original Ego4D data, such as the Ego4D video identification of the clip's source video, among other details. In addition, download tools to acquire and pre-process the video RGB data will also be provided on our website.

**When will the dataset be distributed?**
The full dataset will be made available upon the acceptance of the paper before the camera-ready deadline.

**Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)?** If so, please describe this license and/or ToU, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms or ToU, as well as any fees associated with these restrictions.
EgoSchema will be publicly released under the Ego4D license, which allows direct public use of the video and text data for both research and commercial purposes.

**Have any third parties imposed IP-based or other restrictions on the data associated with the instances?** If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms, as well as any fees associated with these restrictions.
No

**Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?** If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any supporting documentation.
No

## Maintenance

**Who will be supporting/hosting/maintaining the dataset?**
The authors of the paper will be maintaining the dataset, pointers to which will be hosted on github repo https://github.com/egoschema/EgoSchema along with the code for download and preprocessing tool, with the actual data hosted either on Amazon AWS as an S3 bucket or as a google drive folder.

**How can the owner/curator/manager of the dataset be contacted (e.g., email address)?**
We will post the contact information on our website. We will be available through github issues as well as through email.

**Is there an erratum?** If so, please provide a link or other access point.
We will host an erratum on the Github repo in the future, to host any approved errata suggested by the authors or the video research community.

**Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)?** If so, please describe how often, by whom, and how updates will be communicated to users (e.g., mailing list, GitHub)?
Yes, we plan to host an erratum publicly. There are no specific plans for a v2 version, but there does seem plenty oppurtunities for exciting future dataset work based on EgoSchema.

**If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were individuals in question told that their data would be retained for a fixed period of time and then deleted)?** If so, please describe these limits and explain how they will be enforced.
No.

**Will older versions of the dataset continue to be supported/hosted/maintained?** If so, please describe how. If not, please describe how its obsolescence will be communicated to users.
N/A There are no older versions at the current moment. All updates regarding the current version will be communicated via our website.

**If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?** If so, please provide a description. Will these contributions be validated/verified? If so, please describe how. If not, why not? Is there a process for communicating/distributing these contributions to other users? If so, please provide a description.
Contributions will be made possible using standard open-source tools, submitted as pull requests to the relevant GitHub repository. Moreover, we will provide information on how to trace sampled clips back to their original source within the Ego4D dataset. This will enable users to access additional Ego4D data, such as narrations, summaries, and object detections, as applicable.

## Full Prompts

Here are some of the prompts we developed for generating EgoSchema.

### Set A

#### Question prompt

```
Input:
I want you to act as a teacher in the class called "Long-term video
   understanding". I will provide video action narrations and their
   timestamps and you will generate three highly difficult and diverse
   questions for your students about the high-level details in the
   video. You want to test students’ following abilities:

Ability 1: Students’ ability to summarize and compare long parts of the
    video
Ability 2: Students’ ability to compress information from the video
   rather than just listing the actions that happened in the video.
Ability 3: Students’ ability to identify the most important parts of
   the video.

Your questions should not mention any particular timestamps or
   narrations. Remember to make sure the correct answers to your
   questions do not list information from the narrations but compress
   them in a concise conclusion.

Examples of good and difficult questions:
"What is the main action of the video?"
"Why did C do action ...?"

AVOID the following types of questions:
"When ...?"
"How many ...?"
"How much ...?"

When announcing the question please label each question as "Question
   1,2,3: [full question]"


Timestamps and narrations:
0 - C stares at the lamp
1 - C looks around the apartment
3 - C talks to a man
```

```
6 - C walks around the apartment
10 - C talks to a man
11 - C looks around the apartment
14 - a man plays the guitar
20 - a man walks around the apartment
21 - a man plays the guitar
21 - C walks around the apartment
22 - C stares at the window
24 - C looks around the apartment
27 - C talks to a man
29 - C picks chips from the table
32 - C looks around the apartment
36 - C stares at a man
36 - a man plays the guitar
37 - C walks around the apartment
41 - C sits on the sofa
43 - C stares at a man
44 - a man plays the guitar
49 - a man climbs up the sofa
50 - a man plays the guitar
62 - a man climbs down the sofa
63 - a man walks around the apartment
64 - a man plays the guitar
93 - C stares at the table
96 - a man plays a guitar
111 - C stares at the table
112 - a man walks around the apartment
114 - C stares at a man
115 - a man lifts the guitar
118 - a man walks around the apartment
120 - a man places guitar on the sofa
122 - a man walks around the apartment
125 - C stands up
126 - C walks around the apartment
131 - a man picks a coffee maker jug
132 - a man pours coffee in a cup
134 - C stares at a man
140 - C stares at the window
142 - C stares at a man
144 - a man drinks coffee
145 - C looks around the apartment
150 - C walks around the apartment
152 - C talks to a man
156 - C stares at a bench
```

```
158 - C looks around the apartment
164 - C stares at a man
164 - a man talks to C
167 - C looks around the apartment
171 - C follows a man
174 - a man points at the window
175 - C looks around the bedroom
================
Output:
Question 1: What can you deduce about the relationship between C and
   the man, based on their interactions and behaviors throughout the
   video?

Question 2: Identify the primary activity in the apartment and discuss
   its importance/significance. How does it influence the actions and
   atmosphere of the video?

Question 3: Observe the dynamics of the video in terms of changes in
   characters' actions, interactions, and spatial movement. How do
   these shifts contribute to the overall narrative?
```

**Answer prompt**

```
Input:
I want you to act as a teacher in the class called "Long-term video
   understanding." I will provide video action narrations and their
   timestamps and three highly difficult and diverse questions for your
    students about the high-level details in the video. I want you to
   test students' following abilities:

Ability 1: Students' ability to summarize and compare long parts of the
    video
Ability 2: Students' ability to compress information from the video
   rather than just listing the actions that happened in the video.
Ability 3: Students' ability to identify the most important parts of
   the video.

I want you to create a difficult multiple-choice exam that tests above
   student abilities based on the three questions I just provided. Each
    question should have five similar open-ended but short answers, but
    only one should be correct. Make it very difficult for students to
   find the correct answer among all the wrong answers. All answers
   should be closely related to what happens in the video. Make wrong
   answers significantly longer than correct answers. Ensure all of the
    correct answers compress information from narrations them into a
   concise conclusion. Your answers should not mention any particular
   timestamps or narrations.

Do not use letters for the answer choices
Print each correct answer exactly as "Correct answer: [full answer]"
Please print each wrong answer on a new line and print each wrong
   answer as "Wrong answer 1,2,3,4: [full answer]"


Timestamps and narrations:
0 - C stares at the lamp
1 - C looks around the apartment
3 - C talks to a man
6 - C walks around the apartment
10 - C talks to a man
11 - C looks around the apartment
14 - a man plays the guitar
20 - a man walks around the apartment
21 - a man plays the guitar
21 - C walks around the apartment
```

```
22 - C stares at the window
24 - C looks around the apartment
27 - C talks to a man
29 - C picks chips from the table
32 - C looks around the apartment
36 - C stares at a man
36 - a man plays the guitar
37 - C walks around the apartment
41 - C sits on the sofa
43 - C stares at a man
44 - a man plays the guitar
49 - a man climbs up the sofa
50 - a man plays the guitar
62 - a man climbs down the sofa
63 - a man walks around the apartment
64 - a man plays the guitar
93 - C stares at the table
96 - a man plays a guitar
111 - C stares at the table
112 - a man walks around the apartment
114 - C stares at a man
115 - a man lifts the guitar
118 - a man walks around the apartment
120 - a man places guitar on the sofa
122 - a man walks around the apartment
125 - C stands up
126 - C walks around the apartment
131 - a man picks a coffee maker jug
132 - a man pours coffee in a cup
134 - C stares at a man
140 - C stares at the window
142 - C stares at a man
144 - a man drinks coffee
145 - C looks around the apartment
150 - C walks around the apartment
152 - C talks to a man
156 - C stares at a bench
158 - C looks around the apartment
164 - C stares at a man
164 - a man talks to C
167 - C looks around the apartment
171 - C follows a man
174 - a man points at the window
175 - C looks around the bedroom
```

```
Questions:

Question: what can you deduce about the relationship between c and the
   man, based on their interactions and behaviors throughout the video?

Question: identify the primary activity in the apartment and discuss
   its importance/significance. how does it influence the actions and
   atmosphere of the video?

Question: Observe the dynamics of the video in terms of changes in
   characters' actions, interactions, and spatial movement. how do
   these shifts contribute to the overall narrative?
================
Output:
Question 1: What can you deduce about the relationship between C and
   the man, based on their interactions and behaviors throughout the
   video?

Correct answer: They have a casual, friendly relationship.
Wrong answer 1: C and the man are complete strangers and never
   interacted before.
Wrong answer 2: C is deeply in love with the man, and the man
   reciprocates those feelings.
Wrong answer 3: They are business associates discussing a financial
   transaction.
Wrong answer 4: C is an intruder in the man's apartment, and the man is
    trying to get C to leave.

Question 2: Identify the primary activity in the apartment and discuss
   its importance/significance. How does it influence the actions and
   atmosphere of the video?

Correct answer: The primary activity is the man playing the guitar,
   which sets a relaxed environment.
Wrong answer 1: The main activity is C staring at different objects,
   indicating a deep curiosity.
Wrong answer 2: The primary activity is a heated argument between C and
    the man that escalates over time.
Wrong answer 3: The main activity is the man anxiously pacing around
   the apartment, creating a tense atmosphere.
Wrong answer 4: The primary activity is C and the man planning to
   commit a crime together.
```

```
Question 3: Observe the dynamics of the video in terms of changes in
   characters' actions, interactions, and spatial movement. How do
   these shifts contribute to the overall narrative?
Correct Answer: Actions and interactions are casual and relaxed,
   reflecting a comfortable environment.
Wrong Answer A : The video displays a profound sense of conflict and
   tension arising between the characters.
Wrong Answer B : The man is showing C the issues that need fixing in
   the apartment in a professional manner.
Wrong Answer C : Both the characters display an increasingly urgent
   need to solve an issue in the apartment.
Wrong Answer D : C and the man admire and interact with several objects
    in the apartment that look beautiful.
```

## Set B

### Question and answer prompt

```
Input:
I want you to act as a teacher in the class called "Long-term video
   understanding". I will provide video action narrations and their
   timestamps and you will generate three highly difficult and diverse
   questions for your students about the high-level details in the
   video. You want to test students' following abilities:

Ability 1: Students' ability to summarize and compare long parts of the
    video
Ability 2: Students' ability to compress information from the video
   rather than just listing the actions that happened in the video.
Ability 3: Students' ability to identify the most important parts of
   the video.

Your questions should not mention any particular timestamps or
   narrations. Remember to make sure the correct answers to your
   questions do not list information from the narrations but compress
   them in a concise conclusion.

Examples of good and difficult questions:
"What is the main action of the video?"
"Why did C do action ...?"

AVOID the following types of questions:
"When ...?"
"How many ...?"
```

```
"How much ...?"

When announcing the question please label each question as "Question
   1,2,3: [full question]"


Timestamps and narrations:
3 - C holds the cloth in his right hand.
5 - the woman picks a carton from the grocery bag on the floor with her
    right hand.
6 - the woman drops the carton in a cabinet with her left hand.
7 - the woman dips both hands into the grocery bag.
9 - the woman drops a green carton on the floor with her right hand.
12 - C drops the green carton in the cabinet with his right hand.
13 - the woman holds a pack bag in her right hand.
16 - C opens a kitchen cabinet with his left hand.
18 - C removes a cereal pack from the kitchen cabinet with his left
   hand.
19 - C puts the green carton into the kitchen cabinet with his right
   hand.
21 - C closes the kitchen cabinet with his left hand.
24 - the woman removes a plastic from the grocery bag with her right
   hand.
25 - the woman drops the plastic on the floor with her right hand.
33 - C closes a wardrobe with his left hand.
38 - the woman puts a pack into the cabinet with her right hand.
43 - a dog lies down on a bed.
54 - C picks a cloth from the floor with his right hand.
58 - C adjusts the cloth with both hands.
66 - C hangs the cloth on the wall with both hands.
74 - the woman holds a grocery bag in her right hand.
82 - the woman touches her hair with her right hand.
92 - the woman talks with C.
99 - C holds two piece of cloths in both hands.
100 - the woman picks piece of clothes from a bag with both hands.
100 - C adjusts a camera on his head with his right hand.
103 - C drops the two piece of cloths on a couch with his left hand.
109 - C opens a door with his right hand.
110 - C walks into a toilet.
114 - C holds a red towel in his right hand.
116 - the woman talks with C.
118 - the woman scratches her back with her right hand.
118 - the dog walks into the toilet.
120 - C hangs the red towel on the wall with his right hand.
```

```
121 - C holds a towel in his left hand.
129 - the woman holds the towel in her left hand.
129 - the dog walks out of the toilet.
132 - C holds the door with his left hand.
133 - the woman touches her face with her right hand.
148 - C adjusts the camera on his head with his left hand.
149 - C touches his face with his right hand.
150 - the woman holds a bin with both hands.
160 - C opens the door with his right hand.
161 - C walks into the toilet.
163 - the woman walks into the toilet.
166 - the woman drops the bin on the floor with both hands.
168 - the woman adjusts the bin on the floor with both hands.
================
Output:
Question 1: Can you provide a concise summary of the primary activities
    taking place in the video, highlighting similarities between the
   actions of C and the woman?

Question 2: Based on the video, what task seems to be the central focus
    of both C and the woman's actions, and how do they contribute to
   the accomplishment of this task?

Question 3: Identify the three most important segments of the video and
    explain the significance of each in the overall context of the
   actions taking place.
================
Input:
Students are done with the questions. It is time to reveal the correct
   answers. Correctly provide short one-sentence answers to the
   questions you created, and additionally generate 4 very similar but
   wrong answers. Make sure that it will be hard to find the correct
   answer when combined with the wrong ones. Also, ensure that all the
   answers are concise and very close to an actual answer in length.
   Mainly make sure your correct answers do not list information from
   the narrations but compress them in a concise conclusion.

Do not use letters for the answer choices
Print each correct answer exactly as "Correct answer: [full answer]"
Please print each wrong answer on a new line and print each wrong
   answer as "Wrong answer 1,2,3,4: [full answer]"
================
Output:
Question 1:
```

```
Correct answer: C and the woman both interact with various objects to
   organize the space, including placing grocery items and hanging
   cloths.
Wrong answer 1: [Output disregarded]
Wrong answer 2: [Output disregarded]
Wrong answer 3: [Output disregarded]
Wrong answer 4: [Output disregarded]

Question 2:
Correct answer: The central focus is organizing and tidying up the
   space, with C and the woman putting items in their respective places.

Wrong answer 1: [Output disregarded]
Wrong answer 2: [Output disregarded]
Wrong answer 3: [Output disregarded]
Wrong answer 4: [Output disregarded]

Question 3:
Correct answer: Key segments include organizing groceries, adjusting
   household items (e.g., cloths, bins), and interacting with a dog.
Wrong answer 1: [Output disregarded]
Wrong answer 2: [Output disregarded]
Wrong answer 3: [Output disregarded]
Wrong answer 4: [Output disregarded]
```

### Wrong answer prompt

```
Input:
I want you to act as a teacher in the class called "Long-term video
   understanding." I will provide video action narrations and their
   timestamps and three highly difficult and diverse questions for your
    students about the high-level details in the video. I will also
   show the correct answers to the questions. I want you to test
   students' following abilities:

Ability 1: Students' ability to summarize and compare long parts of the
    video
Ability 2: Students' ability to compress information from the video
   rather than just listing the actions that happened in the video.
Ability 3: Students' ability to identify the most important parts of
   the video.

I want you to create a difficult multiple-choice exam that tests above
   student abilities based on the three questions and their correct
   answers I just provided. Each question should have five similar open
   -ended but short answers, but only one should be correct. Make it
   very difficult for students to find the correct answer among all the
    wrong answers. All answers should be closely related to what
   happens in the video. Make wrong answers significantly longer than
   correct answers. Ensure all of the correct answers compress
   information from narrations them into a concise conclusion. Your
   answers should not mention any particular timestamps or narrations.

Do not use letters for the answer choices
Please print each wrong answer on a new line and print each wrong
   answer as "Wrong answer 1,2,3,4: [full answer]"


Timestamps and narrations:
3 - C holds the cloth in his right hand.
5 - the woman picks a carton from the grocery bag on the floor with her
    right hand.
6 - the woman drops the carton in a cabinet with her left hand.
7 - the woman dips both hands into the grocery bag.
9 - the woman drops a green carton on the floor with her right hand.
12 - C drops the green carton in the cabinet with his right hand.
13 - the woman holds a pack bag in her right hand.
16 - C opens a kitchen cabinet with his left hand.
18 - C removes a cereal pack from the kitchen cabinet with his left
```

```
   hand.
19 - C puts the green carton into the kitchen cabinet with his right
   hand.
21 - C closes the kitchen cabinet with his left hand.
24 - the woman removes a plastic from the grocery bag with her right
   hand.
25 - the woman drops the plastic on the floor with her right hand.
33 - C closes a wardrobe with his left hand.
38 - the woman puts a pack into the cabinet with her right hand.
43 - a dog lies down on a bed.
54 - C picks a cloth from the floor with his right hand.
58 - C adjusts the cloth with both hands.
66 - C hangs the cloth on the wall with both hands.
74 - the woman holds a grocery bag in her right hand.
82 - the woman touches her hair with her right hand.
92 - the woman talks with C.
99 - C holds two piece of cloths in both hands.
100 - the woman picks piece of clothes from a bag with both hands.
100 - C adjusts a camera on his head with his right hand.
103 - C drops the two piece of cloths on a couch with his left hand.
109 - C opens a door with his right hand.
110 - C walks into a toilet.
114 - C holds a red towel in his right hand.
116 - the woman talks with C.
118 - the woman scratches her back with her right hand.
118 - the dog walks into the toilet.
120 - C hangs the red towel on the wall with his right hand.
121 - C holds a towel in his left hand.
129 - the woman holds the towel in her left hand.
129 - the dog walks out of the toilet.
132 - C holds the door with his left hand.
133 - the woman touches her face with her right hand.
148 - C adjusts the camera on his head with his left hand.
149 - C touches his face with his right hand.
150 - the woman holds a bin with both hands.
160 - C opens the door with his right hand.
161 - C walks into the toilet.
163 - the woman walks into the toilet.
166 - the woman drops the bin on the floor with both hands.
168 - the woman adjusts the bin on the floor with both hands.

Questions:

Question: Can you provide a concise summary of the primary activities
```

```
   taking place in the video, highlighting similarities between the
   actions of c and the woman?
Correct answer: c and the woman both interact with various objects to
   organize the space, including placing grocery items and hanging
   cloths.

Question: Based on the video, what task seems to be the central focus
   of both c and the woman's actions, and how do they contribute to the
    accomplishment of this task?
Correct answer: the central focus is organizing and tidying up the
   space, with c and the woman putting items in their respective places.


Question: Identify the three most important segments of the video and
   explain the significance of each in the overall context of the
   actions taking place?
Correct answer: key segments include organizing groceries, adjusting
   household items (e.g., cloths, bins), and interacting with a dog.
================
Output:
Question 1: Can you provide a concise summary of the primary activities
    taking place in the video, highlighting similarities between the
   actions of c and the woman?

Correct answer: C and the woman both interact with various objects to
   organize the space, including placing grocery items and hanging
   cloths.
Wrong answer 1: C and the woman engage in lengthy conversations about
   their day while occasionally placing groceries in cabinets and
   adjusting cloths on walls.
Wrong answer 2: C and the woman focus on preparing an elaborate dinner,
    flipping through recipe books and chopping vegetables.
Wrong answer 3: Both C and the woman predominantly pay attention to the
    dog's well-being, providing it with food and water and playing with
    it.
Wrong answer 4: The video revolves around C and the woman working on a
   DIY project, with tools scattered around the room.

Question 2: Based on the video, what task seems to be the central focus
    of both c and the woman's actions, and how do they contribute to
   the accomplishment of this task?

Correct answer: The central focus is organizing and tidying up the
   space, with C and the woman putting items in their respective places.
```

```
Wrong answer 1: C and the woman are preoccupied with an ongoing debate,
    their actions aimlessly directed towards unrelated tasks.
Wrong answer 2: They are both primarily absorbed in their individual
   tasks, occasionally interacting with each other but with no clear
   common goal.
Wrong answer 3: The central focus of their actions is providing care
   and attention to the dog, and their actions revolve around its needs.

Wrong answer 4: The video primarily depicts a disagreement between C
   and the woman, with actions centered around this conflict.

Question 3: Identify the three most important segments of the video and
    explain the significance of each in the overall context of the
   actions taking place?

Correct answer: Key segments include organizing groceries, adjusting
   household items (e.g., cloths, bins), and interacting with a dog.
Wrong answer 1: The main segments involve discussing a movie, checking
   on an overheating stove, and attempting to catch a rodent in the
   room.
Wrong answer 2: The most important segments consist of C and the woman
   eating a meal, participating in a board game, and reading books.
Wrong answer 3: Three primary segments include C and the woman debating
    household chores, searching for lost items, and tending to a crying
    baby.
Wrong answer 4: Key segments include discussing an upcoming event,
   taking turns answering a phone call, and checking updates from an
   ongoing sports game.
```

## C.1 Our clip length and narration density choice

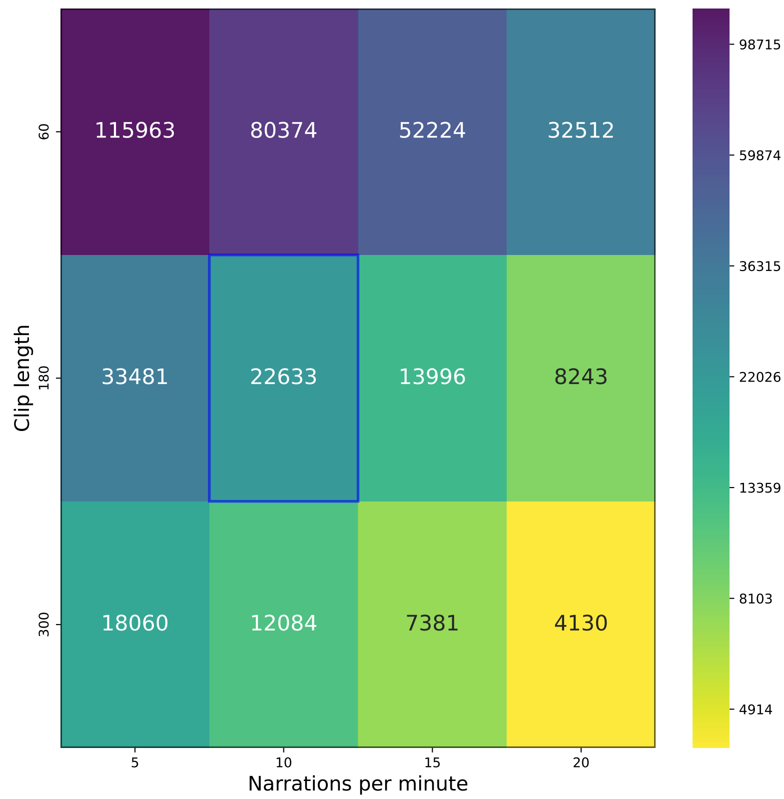



Figure A.1: **Heatmap of number of viable clips over a range of clip length and narration density.** There are only a few viable options that offer some degree of balance between the number of clips and the number of narration in the clip. One potential selection is to utilize 3-minute clips with a density of 5 narrations per minute, although this choice bears the significant disadvantage of potentially including clips with an insufficient volume of narration data to generate high-quality results. Another possible choice is to use 1-minute clips with a density of 20 narrations per minute, yet this option carries the drawback of the clips being too brief for the dataset to be very long-term. Hence, we choose 3-minute clips with a narration density of 10 narrations per minute as it offers a satisfactory balance between the number of narrations and clip length for generating EgoSchema.

## C.2 Human curation

Our research utilized the services of a third part company (not MTurk), for specifically training annotators to ensure quality. The process involved two distinct annotation procedures: data curation and human accuracy testing.

## Curation

Generated data curation was performed by Quantigo employees. These curators were responsible for ensuring that the released EgoSchema dataset is tge highest high quality possible. Here is the exact instructions that was provided to annotators:

```
The annotation we need is to say that the Question-correct answer-wrong
    answer set (the whole set) is good if all these three conditions
   pass:

(Condition A) Question is Answerable: The question can be answered from
    the video and requires more than just a second of video to answer (
   so, if the answer is not present in the video or, if the answer can
   be formed with just a few frames (less than say, a second) then it
   fails this condition).

(Condition B) The Marked Correct Answer is correct: The ""correct
   answer"" is more the correct answer to the question

(Condition C) The Marked Wrong Answers are wrong: All 4 ""wrong answers
   "" are less correct than the ""correct answer"" (So for example, if
   a wrong answer is not completely false, but simply does not contain
   all the information that the ""correct answer"" does, then it is
   still a fine ""wrong answer"") IF even one of the marked answer is
   correct, the set should be labeled as bad.

(Condition D) The question is actually long-term: This is a very very
   important condition. We want the certificate for the question to be
   at least 30 seconds minimum. If the certificate is non-contiguous (
   ie. 5 seconds at one place, 20 seconds at another, and 15 more
   seconds at a third place) the sum of lengths of all the sub-
   certificates together should be more than 30 seconds. Another
   example is, if a question can be answered simply from a few frames
   of the video, the certificate is small (and less than 30 seconds)
   and hence would fail this condition. Additional detials on how to
   handle certificate edge cases are provided in the annotator training
    through examples.

(Condition E) Avoid Boring Questions: Questions that ask about the
   frequency of something ("How many times..") fail this condition.

If any of these five conditions fail we want the whole set (Question /
   Correct Answer / Wrong Answer) marked bad.

Optional:
```

```
Since GOOD sets are so rare, in cases where it seems that a set is good
    but a small part of the above five conditions is not being met or,
   if one/two words were different this can be a good set, please label
    as MAYBE and we will fix it in the second round. We expect, Good/
   Bad to be about 97% of data and Maybe to be not more than 3%.

Extended notes:
1. In our experience, the wrong answers are made such that they differ
   from the correct answer in small but crucially meaningful ways.
   There are many cases where a careless reading of the wrong answer
   might make it seem that it is correct but upon careful inspection,
   it will become clear that something about the wrong answer indeed
   makes it wrong. While this is common, there are indeed cases where
   the wrong answer is also just as correct as the correct answer. In
   such cases, the wrong answer fails condition C, and the set is bad.

2. Roughly speaking, we expect about 20-25% of the questions that we
   have provided to be found as good. However, this is not necessary
   and the %age can be smaller or larger depending on each three-minute
    clip.

Edge Cases:
1. If the asked question has multiple answers and at least one of them
   aligns with the correct answer while none of them align with any of
   the other wrong answers, then provided that the top 5 conditions are
    met, we can mark the set as good.
2. If two questions are very similar (even within different clips) and
   both are GOOD, only choose one as GOOD and reject the other one with
    a comment mentioning this. We do not expect this to happen more
   than 1 or 2 times in a 100.
3. There might be more such edge cases, please feel free to contact me
   in such cases and we can expland.
```

## C.3 Benchmarking details

### Violet

Violet is a video language model comprised of a visual encoder, text encoder, and multimodal transformer pretrained on a variety of masked visual modeling tasks ranging from simple ones such as RBG pixel values up to more high levels ones such as spatially focussed image features. It performs competitively on a variety of video-language tasks such as Video-QA and Video-Text Retrieval. We evaluate one pre-trained model and 3 models finetuned on lsmdc-mc, msrvtt-qa, and msrvtt-retrieval. We evaluate using both 5 frames and 75 frames and choose the model with the best overall accuracy.

## mPLUG-Owl

By default, mPLUG-Owl does not possess inherent capabilities for direct video question answering. As such, we undertook several experiments to adapt it to our required format. One approach involved inputting all answer choices in the form of a shuffled test. However, this resulted in a bias towards selecting the first option in most cases. For another approach, FrozenBiLM offered a methodology for frozen zero-shot models to operate in the context of multiple-choice video question answering, which inspired us to adapt this methodology for mPLUG-Owl. As mPLUG-Owl utilizes word-level tokenization, we could extract the confidence score for each generated token, particularly the 'Yes' token. We recorded the 'Yes' token confidence score for each answer option. In instances where the 'Yes' token was absent, we assigned the confidence score as zero, though empirically, in most cases, the model output was positive and contained the 'Yes' token. Ultimately, we selected the answer option with the highest 'Yes' confidence score as the model output given the question. In scenarios where multiple options scored the same highest confidence for the 'Yes' token, we randomly selected the answer from these top-scoring options. It should be noted that mPlug-Owl was originally trained to process a single image, and its capacity to handle additional frames is an emergent ability that has not been thoroughly tested to date."

## InternVideo

The two most closely aligned formats supported by InternVideo are open-ended Video Question Answering and Zero-shot Multiple Choice tasks. In the case of open-ended Video Question Answering, the task is to predict the answer to a question posed within a video. However, due to the restricted vocabulary of open-ended answers in open-ended Video Questions Answering, we decided to formulate EgoSchema within the context of a Zero-shot Multiple Choice task. This task aims to identify the correct answer from a set of given options, without the inclusion of a question. InternVideo has provided finetuned weights for two datasets: MSRVTT and LSMDC. We selected the model finetuned on MSRVTT because it shares greater contextual similarity with EgoSchema.

## Human

To conduct human benchmarking, we engaged a distinct team of ten employees within the same data annotation company to carry out human benchmarking on our dataset. The answers were randomized and presented in the form of a test. The following are the precise instructions provided to the annotators:

```
- Setting 1: Unlimited setting -- The goal is to get answers as
   accurately as possible without worrying about time.
```

```
- Setting 2: 1 minute timed setting -- In this case, the test taker (
   annotator) has only 1 minute to spend per question (including
   watching video/reading text/everything). If they do not have the
   answer, just guess the best based on their intuition and move on.

- Setting 3: 3 minutes timed setting-- Same as above but with 3 min
   instead of just one.

- Setting 4: Video -> Text setting -- In this case, the taker is not
   allowed to read the text before looking at the video and at the
   video after reading the text. In other words, the test taker can
   spend as much time as they want to look at the video first and then
   must move on to answering the question. They cannot go back to the
   video once they start reading the text. This is an untimed setting
   -- they can take as much time as they want per question.

- Setting 5: 180 frames setting -- This is the same as the untimed
   setting except the annotator has access to only 1 frame per second (
   ie the video feels like a GIF with one frame per second instead of
   the usual 30 frames per second) -- each video is still 3 minutes
   long, but it feels more jittery. All instructions remain the same as
    in an untimed setting.
```

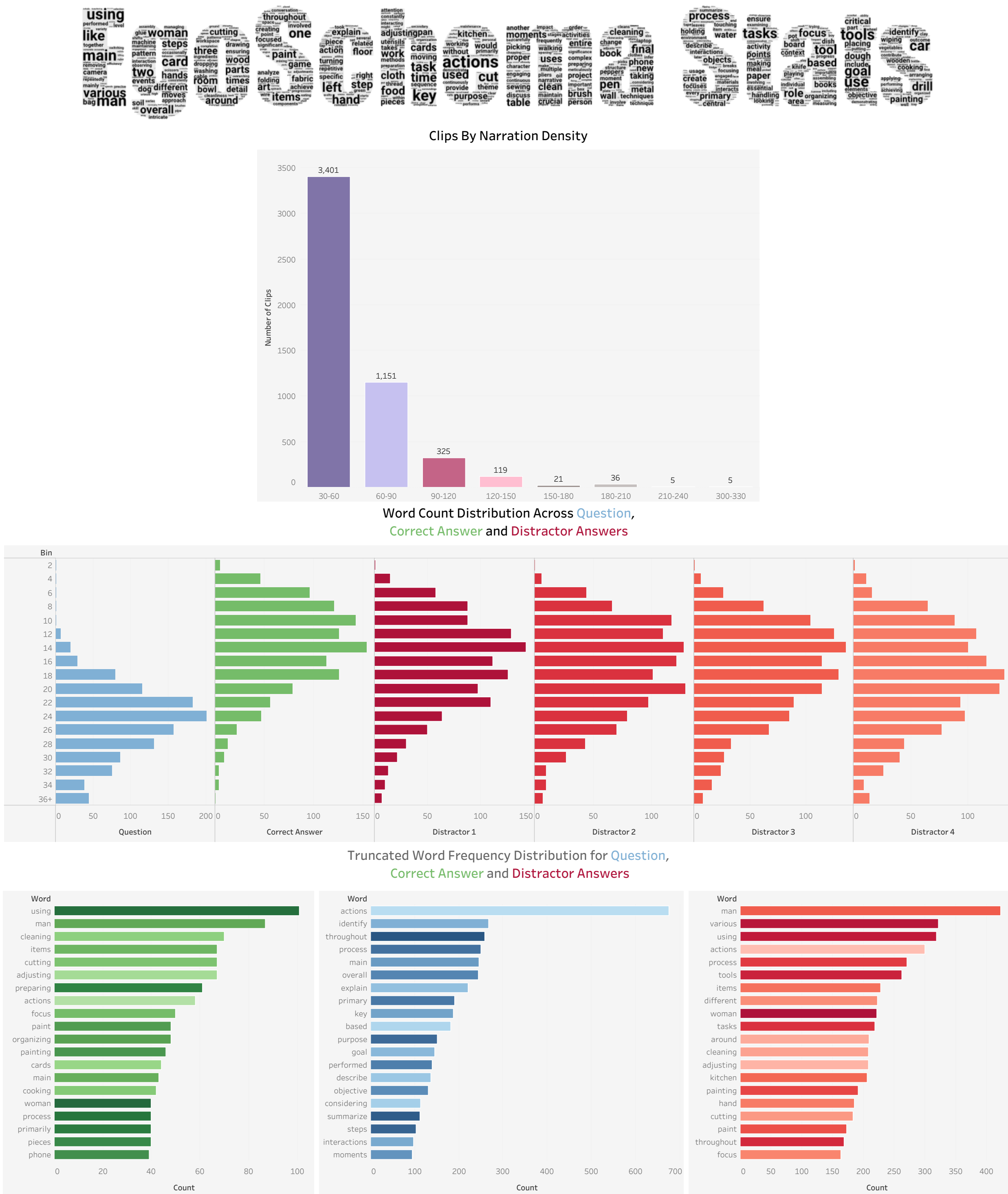



Figure A.2: Interactive Version of these statistics visualizations can be found at the statistics page on our website.