Vision and Language Understanding Through Generative Modeling

SETH PARK

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2023-202
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2023/EECS-2023-202.html

August 8, 2023

Vision and Language Understanding Through Generative Modeling

By

Dong Huk S Park

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 Trevor Darrell, Chair
Professor Dan Klein
Associate Professor David Bamman

Summer 2023

Vision and Language Understanding Through Generative Modeling

Abstract

Vision and Language Understanding Through Generative Modeling

by

Dong Huk S Park

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Trevor Darrell, Chair

Language is such a powerful representation for capturing the knowledge and information about our world. It excels at expressing discrete concepts such as objects and their attributes, the relationships between them in a very compact manner all due to its extremely high level of abstraction. Language is the primary means by which we communicate, comprehend, and express our thoughts and ideas, and it lies at the very core of human intelligence. With the advent of powerful generative models, machines also have begun to comprehend and generate natural language with notable fluency and creativity. However, they lack “grounding”—a direct tie to the visual world. Vision plays a pivotal role in our comprehension and production of language. When we describe a scene, understand instructions, or engage in a dialogue, visual context significantly aids our interpretation and generation of language. This highlights the need for integrating vision for generative modeling.

Chapter 1 and 2 delve into image-to-text domain, spotlighting the importance of a multimodal approach for text generation. In Chapter 1, we explore how generating textual rationales with attention visualizations can enhance model transparency for visual question answering. In Chapter 2, we build generative models that abandon traditional left-to-right sequencing in favor of an unsupervised technique to determine optimal generation orders. Chapter 3 and 4 shift the focus to text-to-image generation. In Chapter 3, we introduce a training-free framework that combines linguistic cues with reference images, allowing for controllable image synthesis using denoising diffusion probabilistic models. Lastly, Chapter 4 emphasizes the importance of preserving object shapes in text-based image editing, proposing a unique mechanism that augments text-to-image models to be more faithful to input masks and text prompts.

To my parents, Jung Duk Gu and Kyung Deuk Park.

Contents

List of Figures

List of Tables

Acknowledgments

My time at Berkeley stands out as the most unforgettable period of my life. Having lived nearly a decade as an undergraduate and graduate student here, I will always remember Berkeley as a place full of cherished memories. Thank you all to the people around me who gifted me with such memories.

Thank you to my advisor, Trevor Darrell, for guiding me through this long journey. You have been a great advisor, supporting and motivating me to freely pursue ideas that I am most excited about. Thank you for your guidance and encouragement. I have truly enjoyed my time here as your student.

Thank you to Marcus and Anna Rohrbach. Marcus, I still remember the time I reached out to you as an undergrad to do research at the group. Thank you for giving me the opportunity to discover what I am most passionate about. Without your help, I would have not even considered embarking on this journey as a PhD student. And Anna, thank you for being my unofficial co-advisor. You have been greatly influential to every work I have done and I appreciate your mentorship and support. This dissertation would have not been possible without you.

I would also like to extend my gratitude to the professors who served on my committees: Hany Farid, for helping me with my qualifying examination, and Dan Klein and David Bamman for helping me with both by qualifying examination and dissertation. Thank you all for the valuable feedback and thoughtful insights.

I have been fortunate to have so many friends, colleagues, and collaborators at Berkeley and other places: Eric Tzeng, Evan Shelhamer, Lisa Anne Hendricks, Samaneh Azadi, Parsa Mahmoudieh, Sayna Ebrahimi, Devin Guillory, Xihui Liu, Amir Bar, Dave Epstein, Suzie Petryk, and Grace Luo. From brainstorming research ideas to sharing small talks and jokes, being with you all has made my experience here much more enjoyable and memorable.

Thank you to the most talented team at Pinterest: Dmitry Kislyuk, Andrew Zhai, Rex Wu, Josh Beal, Eric Kim, David Xue, and Nadia Fawaz. Working with you all has taught me things that I would have not been able to learn from an academic setting, and it was a tremendoulsy valuable experience to see how computer vision can be applied to real-world use cases that impact millions of people.

I would also like to show gratitude toward my close friends outside school: Min Gu Jo, Dong Hyeon Ko, Daniel Lim, Erin Choi, Sharon Lee, Jen Won, DJ Min, Jin Guen Lee, Jae Guen Lee, Kyung Mo Kang, Jason Seung Soo Lee, and Jin Woo Hong. Thank you for all the fun moments that made my frustrations go away. You all have provided me with great emotional support, laughter, and encouragement.

Special mention goes to Ah Young Kim, who has always been there for me, encouraging me through highs and lows, and being a constant pillar of strength and support in both good times and bad. Thanks for being my biggest supporter, warmest comforter, and best friend.

Finally, I would like to thank my mom, dad, and sister for their unconditional love and support. Mom, I can never thank you enough for what you have done for me your entire life. Thank you for raising me to be the person I am today. Dad, thank you for providing me with everything I needed to succeed in my life. I only made it here because of your enduring love and dedication. I really miss you. This thesis is dedicated to you.

To the countless people, both named and unnamed, who helped me throughout this journey, I am truly grateful. Without all of you, I would not have made this far. It's the little gestures, conversations, and moments shared that have made all the difference. From the bottom of my heart, thank you for being there with me every step of the way.

Chapter 1

Introduction

Over the last decade, we have seen great advancements of larger and enhanced language models. Important technical developments such as sequence-to-sequence learning [9–12] and generative modeling [13] driven by Transformers [14] have been at the heart of these underlying advances. When trained on a massive corpus of texts sourced from the web, these models are capable of comprehending and generating natural language like humans do, demonstrating unprecedented capabilities in solving a variety of natural language processing (NLP) tasks such as document summarization, machine translation, code completion, and question answering.

While these models have demonstrated remarkable prowess in processing text, there are fundamental challenges they face in truly grasping human intelligence. One significant limitation lies in the fact that these generative models primarily learn from textual data without a direct connection to the physical world. They lack what is known as "grounding" or a visual understanding of the context in which language exists. Language, after all, is intrinsically tied to our sensory experiences and our interaction with the visual environment.

Joint modeling of images and texts has been actively explored in the form of multimodal generative models. They can be largely categorized into image-to-text models [15–20] where models are mainly optimized by learning how to generate texts that are coherent to the conditioned visual information, and text-to-image models in which the common training objective is to generate images that are consistent with the input texts [21–27]. These models are trained on large-scale multimodal datasets collected from the web which contain arbitrary text and image pairs. The scale and quality of these datasets are integral to endowing them with remarkable generalization capabilities, often known as in-context or zero-shot/few-shot learning. By seamlessly bridging the gap between vision and language, these models are enabling more intuitive interfaces for interacting with AI agents, fostering richer content generation, and paving the way for innovative solutions that enrich our lives.

In light of these progresses, this thesis aims to address some of the limitations multimodal generative models have, and explores methodologies to improve them. More concretely, this thesis covers image-to-text models in the domain of visual question answering (VQA), activity recognition, and image captioning, and text-to-image models for image synthesis and shape-guided editing.

In Chapter 2, we propose a multimodal approach to make VQA systems more explainable where the models provide joint textual rationale and attention visualization, and argue that the two modalities provide complementary explanatory strengths. We further demonstrate that training with the textual explanations not only yields better textual justification models, but also models that better localize the evidence that supports the decision.

Chapter 3 introduces an unsupervised parallelizable learner that discovers high-quality text generation orders purely from training data, deviating away from the conventional left-to-right ordering. The learner contains an encoder network and decoder language model that perform variational inference with autoregressive orders (represented as permutation matrices) as latent variables. The corresponding ELBO is not differentiable, so we develop a practical algorithm for end-to-end optimization using policy gradients. We demonstrate the efficacy of our method on diverse tasks such as code generation, machine translation, and image captioning.

In Chapter 4, we shift our focus to text-to-image task and showcase a novel unified framework for semantic diffusion guidance, which allows either language or image guidance, or both to be injected into a pretrained unconditional diffusion model for image synthesis. By using the gradient of image-text or image matching scores, we demonstrate that an unconditional image diffusion model can be repurposed to become text and/or image conditional without any type of re-training. Moreover, the proposed approach can be applied to datasets without associated text annotations which makes it easier to be used as a drop-in solution

Chapter 5 examines a rather different problem from Chapter 4 where we identify a key issue in existing text-to-image diffusion models. Namely, when manipulating an object they often ignore the shape of the object and generate content that is incorrectly scaled, cut off, or replaced with background content. We propose a training-free method, Shape-Guided Diffusion, that modifies pretrained diffusion models to be sensitive to shape input. We use a novel Inside-Outside Attention mechanism during the inversion and generation process designates which spatial region is the object (inside) vs. background (outside) then associates edits specified by text prompts to the correct region.

Finally, in Chapter 6, we summarize the findings of this thesis, and discuss possible extensions and future avenues for further research.

Chapter 2

Multimodal Explanations: Justifying Decisions and Pointing to the Evidence

2.1 Introduction

Explaining decisions is an integral part of human communication, understanding, and learning, and humans naturally provide both deictic (pointing) and textual modalities in a typical explanation. We aim to build deep learning models that also are able to explain their decisions with similar fluency in both visual and textual modalities. Previous machine learning methods for explanation were able to provide a text-only explanation conditioned on an image in context of a task, or were able to visualize active intermediate units in a deep network performing a task, but were unable to provide explanatory text grounded in an image.

We propose a new model which can jointly generate visual and textual explanations, using an attention mask to localize salient regions when generating textual rationales. We argue that to train effective models, measure the quality of the generated explanations, compare with other methods, and understand when methods will generalize, it is important to have access to ground truth human explanations. Unfortunately, there is a dearth of datasets which include examples of how humans justify specific decisions. Thus, we collect two new datasets, ACT-X and VQA-X, which allow us to train and evaluate our novel model, which we call the Pointing and Justification Explanation (PJ-X) model. PJ-X is explicitly multimodal: it incorporates an explanatory attention step, which allows our model to both visually point to the evidence and justify a model decision with text.

To illustrate the utility of multimodal explanations, consider Figure 2.1. In both examples, the question “Is this a healthy meal?” is asked, and the PJ-X model correctly answers either “no” or “yes” depending on the visual input. To justify why

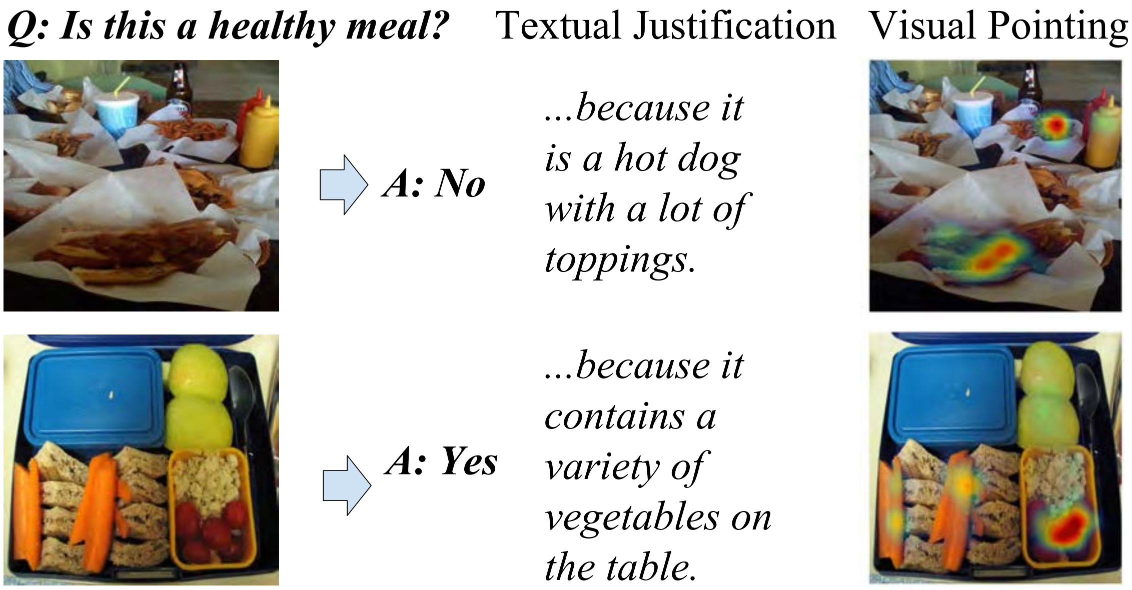

Q: Is this a healthy meal?
Textual Justification
Visual Pointing
A: No
...because it is a hot dog with a lot of toppings.
A: Yes
...because it contains a variety of vegetables on the table.

Figure 2.1: For a given question and an image, our Pointing and Justification Explanation (PJ-X) model predicts the answer and *multimodal* explanations which both point to the visual evidence for a decision and provide textual justifications. We show that considering multimodal explanations results in better explanations as visual and textual components complement each other.

the image is not healthy, the generated textual justification mentions the kinds of unhealthy food in the image ("hot dog" and "toppings"). In addition to mentioning the unhealthy food, our model is able to *point* to the hot dog in the image. Likewise, to justify why the image on the right is healthy, the textual explanation mentions "vegetables". The PJ-X model then points to the vegetables, which are mentioned in the textual explanation, but not other items in the image, such as the bread.

We propose VQA and activity recognition as testbeds for studying explanations because they are challenging and important visual tasks which have interesting properties for explanation. VQA is a widely studied multimodal task that requires visual and textual understanding as well as commonsense knowledge. The newly collected VQA v2 dataset [28] includes complementary pairs of questions and answers. Complementary VQA pairs ask the same question of two semantically similar images which have different answers. As the two images are semantically similar, VQA models must employ finegrained reasoning to answer the question correctly. Not only is this an interesting and useful setting for measuring overall VQA performance, but it is also interesting when studying explanations. By comparing explanations from complementary pairs, we can more easily determine whether our explanations focus on the important factors for making a decision.

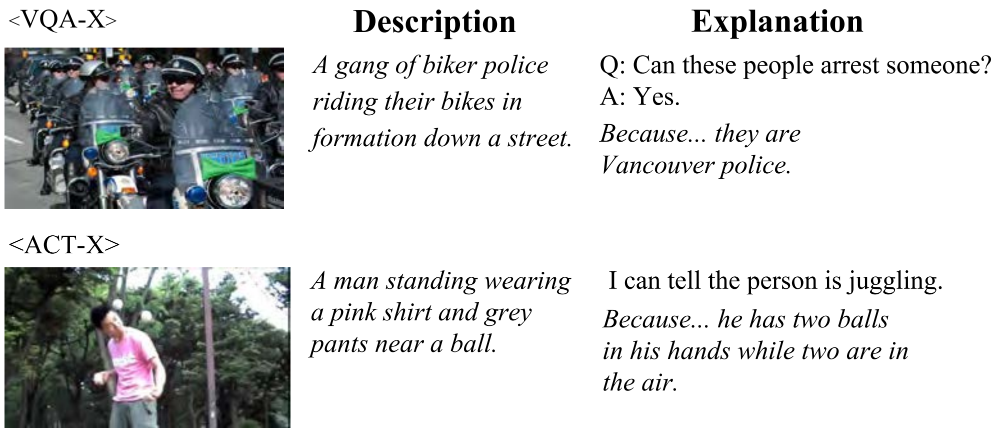
Additionally, we collect annotations for activity recognition using the MPII Human Pose (MHP) dataset [2]. Activity recognition in still images relies on a variety of cues, such as pose, global context, and the interaction between humans and objects. Though a recognition model can potentially classify an activity correctly, it is not capable of indicating which factors influence the decision process. Furthermore, classifying specific activities requires understanding finegrained differences (e.g., "road biking" and "mountain biking" include similar objects like "bike" and "helmet," but road biking occurs on a road whereas mountain biking occurs on a mountain path). Such finegrained differences are interesting yet difficult to capture when explaining neural network decisions.

In sum, we present VQA-X and ACT-X, two novel datasets of human annotated multimodal explanations for activity recognition and visual question answering. These datasets allow us to train the Pointing and Justification (PJ-X) model which goes beyond current visual explanation systems by producing *multimodal* explanations, justifying the predicted answer post-hoc through visual pointing and textual justification. Our datasets also allow us to effectively evaluate explanation models, and we show that the PJ-X model outperforms strong baselines. Importantly, by generating multimodal explanations, we outperform models which only produce visual or textual explanations.

2.2 Related Work

Explanations. Early textual explanation models span a variety of applications (e.g., medical [29] and feedback for teaching programs [30–32]). More recently, [33] developed a deep network to generate natural language justifications of a fine-grained classifier. Unlike our model, it does not provide multimodal explanations and is not trained on reference human explanations as no such dataset existed.

Many works have proposed methods to explain decisions visually. Some methods find discriminative visual patches [34, 35] whereas others aim to understand what specific neurons represent [36–38]. Perhaps the most prevalent form of visual explanation rely on producing heat maps/attention maps which indicate which region of an image is most important for a decision [39–42]. Our PJ-X model points to visual evidence via an attention mechanism [9] which conveys knowledge about what evidence is important without requiring domain knowledge to understand. Explanation systems can either be *introspective* systems, which are designed to reflect the inner workings and decision processes of deep networks, or *justification* systems, which are designed to communicate which visual evidence supports a decision. In this paradigm, models



A gang of biker police riding their bikes in formation down a street.

Q: Can these people arrest someone?
A: Yes.
Because... they are Vancouver police.

A man standing wearing a pink shirt and grey pants near a ball.

I can tell the person is juggling.
Because... he has two balls in his hands while two are in the air.

Figure 2.2: In comparison to descriptions, our VQA-X explanations focus on the evidence that pertains to the *question and answer* instead of generally describing the scene. For ACT-X, our explanations are task specific whereas descriptions are more generic. Images are from [1] and [2].

like [43] which highlight discriminative image attributes without attempting to model the classifiers reasoning process are considered justification explanations, whereas models like [36, 38, 41] which aim to illuminate the inner reasoning process of deep networks are considered introspective explanations. We argue that both are useful. Though justifications would not be necessarily helpful for an engineer debugging an AI component, we assert justification is a core AI problem in and of itself: not only is it an AI challenge to answer "is this image a calico cat," but also we claim it is a foundational AI challenge to answer "why would one say this is an image of a calico cat." Though we train justification systems in this work, the data we have collected could be used to understand how well introspective explanations align with our human annotated justifications.

Prior work investigated how well generated visual explanations align with human gaze [44]. However, when answering a question, humans do not always look at image regions which are necessary to explain a decision. For example, given the question "What is the restaurant's name?" human gaze might capture other buildings before settling on the restaurant. When we collect annotations, annotators view the entire image and point to the most relevant visual evidence for making a decision. Furthermore, visual explanations are collected in conjunction with textual explanations to build and evaluate multimodal explanation models.

Visual Question Answering and Attention. Initial approaches to VQA used full-frame representations [45], but most recent approaches use some form of spatial attention [46–53]. We base our method on [52], the winner of VQA 2016 challenge,

Dataset	**Split**	#Imgs	Q/A	U.Q.	U.A.	Expl.	Avg. #w	Vocab	Comple.	V.Ann.
VQA-X	Train	24876	29459	12942	1147	31536	8.56	12412	6050	–
	Val	1431	1459	813	246	4377	8.89	4325	240	3000
	Test	1921	1968	898	272	5904	8.94	4861	510	3000
	Total	28180	32886	13921	1236	41817	8.64	14106	6800	6000
ACT-X	Train	12607	–	–	397	37821	13.96	12377	–	–
	Val	1802	–	–	295	5406	13.91	4802	–	3000
	Test	3621	–	–	379	10863	13.96	6856	–	3000
	Total	18030	–	–	397	54090	13.95	14588	–	6000

Table 2.1: Dataset statistics for VQA-X (top) and ACT-X (bottom). Q/A = Question/Answer pairs, U.Q. = Unique questions, U.A. = Unique answers, Expl. = Explanations, Avg. #w = Average number of words, Comple. = Complementary pairs, V.Ann. = Visual annotations.

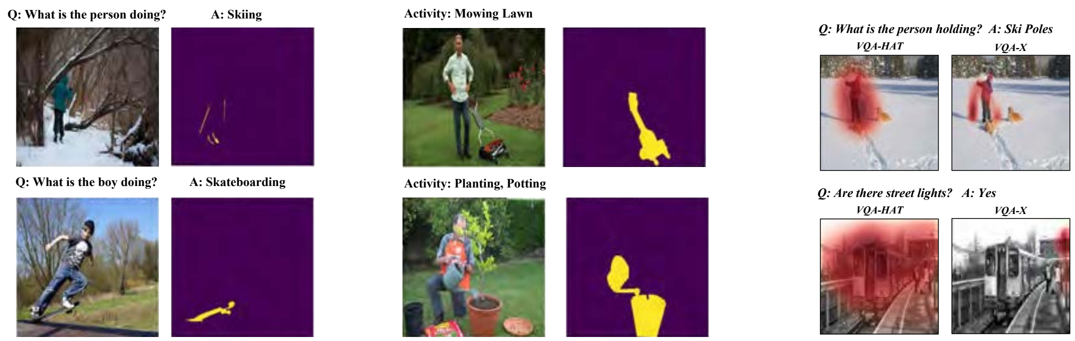
but use an element-wise product as opposed to compact bilinear pooling. [53] also explore the element-wise product for VQA, but [53] improves performance by applying hyperbolic tangent (TanH) after the multimodal pooling whereas we improve by applying signed square-root and L2 normalization.

Activity Recognition. Recent work on activity recognition in still images relies on a variety of cues, such as pose and global context [54–56]. Specifically, [54] considers additional image regions and [55] considers a global image feature in addition to the region where an activity occurs. Generally, works on the MPII Human Activities dataset provide the ground truth location of a human at test time [54]. In contrast, we consider a more realistic scenario and do not provide the ground truth location of humans at test time. Our model relies on attention to focus on important parts of an image for classification and explanation.

2.3 Multimodal Explanations

We propose multimodal explanation tasks with visual and textual components, defined on both visual question answering and activity recognition testbeds. To train and evaluate models for this task we collect two multimodal explanation datasets: Visual Question Answering Explanation (VQA-X) and Activity Explanation (ACT-X) (see Table 2.1 for a summary). For each dataset we collect textual explanations (see Figure 2.2) and visual explanations (see Figure 2.3) from human annotators.

VQA Explanation Dataset (VQA-X). The Visual Question Answering (VQA) dataset [57] contains open-ended questions about images which require understanding


Q: What is the person doing?
A: Skiing
Q: What is the boy doing?
A: Skateboarding
Activity: Mowing Lawn
Activity: Planting, Potting
Q: What is the person holding? A: Ski Poles
VQA-HAT
VQA-X
Q: Are there street lights? A: Yes
VQA-HAT
VQA-X

((a)) Example annotations collected on VQA-X dataset.

((b)) Example annotations collected on ACT-X dataset.

((c)) VQA-HAT vs VQA-X.

Figure 2.3: Human annotated visual explanations. The visual evidence that justifies the answer is segmented in yellow. Images are from [1] and [2].

vision, language, and commonsense knowledge to answer. VQA consists of approximately 200K MSCOCO images [1], with 3 questions per image and 10 answers per question.

Many questions in VQA are of the sort: "What color is the banana?" which is difficult to explain because it requires explaining a fundamental visual property: color. To provide textual explanations for questions that go beyond such trivial cases, we consider the annotations collected in [58] which say how old a human must be to answer a question. We find that questions which require humans to be of age 9 or higher are generally interesting to explain.

Additionally, we consider complementary pairs from the VQA v2 dataset [59]. Complementary pairs consist of a question and two similar images which give two different answers. Complementary pairs are particularly interesting for the explanation task because they allow us to understand whether explanatory models name the correct evidence based on image content, or just memorize which content to consider based off specific question types. We collect one textual explanation for QA pairs in the training set and three textual explanations for test/val set.

Action Explanation Dataset (ACT-X). The MPII Human Pose (MHP) dataset [2] contains 25K images extracted from Youtube videos. From the MHP dataset, we select all images that pertain to 397 activities, resulting in 18, 030 images total. For each image we collect three explanations. During data annotation, we ask the annotators to complete the sentence "I can tell the person is doing (action) because.." where the action is the ground truth activity label. We also ask them to use at least

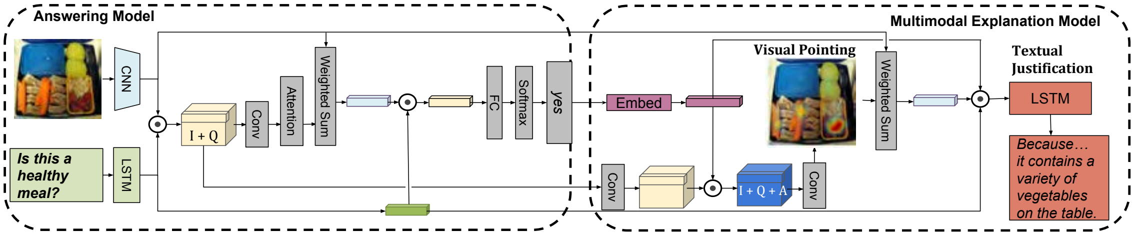


Figure 2.4: Our Pointing and Justification (PJ-X) architecture generates a multimodal explanation which includes textual justification ("it contains a variety of vegetables on the table") and points to the visual evidence.

10 words and avoid mentioning the activity class in the sentence. MHP dataset also comes with sentence descriptions provided by [60].

Ground truth for pointing. In addition to textual justifications, we collect visual explanations from humans for both VQA-X and ACT-X datasets in order to evaluate how well the attention of our model corresponds to where humans think the evidence for the answer is. Human-annotated visual explanations are collected via Amazon Mechanical Turk where we use the segmentation UI interface from the OpenSurfaces Project [61]. Annotators are provided with an image and an answer (question and answer pair for VQA-X, class label for ACT-X). They are asked to segment objects and/or regions that most prominently justify the answer. Some examples can be seen in Figure 2.3.

Comparing with VQA-HAT. A thorough comparison between our dataset and VQA-HAT dataset from [44] is currently not viable because the two datasets have different splits and the overlap is small. However, we present qualitative comparison in Figure 2.3(c). In the first row, our VQA-X annotation has a finer granularity since it segments out the objects in interest more accurately than the VQA-HAT annotation. In the second row, our annotation contains less extraneous information than the VQA-HAT annotation. Since the VQA-HAT annotations are collected by having humans "unblur" the images, they can introduce noise when irrelevant regions are uncovered.

2.4 Pointing and Justification Model (PJ-X)

We implement a multimodal explanation system that justifies a decision with natural language and points to the evidence. Our Pointing and Justification Model (PJ-X) is explicitly trained for these two tasks and relies on natural language justifications and the classification labels as the only supervision. The PJ-X model learns to point in a latent way using an attention mechanism [9] which allows it to focus on a spatial subset of the visual representation.

We first predict the answer given an image and question using the answering model. Then given the answer, question, and image, we generate visual and textual explanations with the multimodal explanation model. An overview of our model is presented in Figure 2.4.

Answering model. In visual question answering the goal is to predict an answer given a question and an image. For activity recognition we do not have an explicit question. Thus, we ignore the question which is equivalent to setting the question representation to $f^Q(Q) = 1$, a vector of ones.

We base our answering model on the overall architecture from the MCB model [52], but replace the MCB unit with a simpler element-wise multiplication $\odot$ to pool multimodal features. This leads to similar performance, but trains faster.

In detail, we extract spatial image features $f^I(I, n, m)$ from the last convolutional layer of ResNet-152 followed by 1×1 convolutions $(\bar{f}^I)$ giving a $2048 \times N \times M$ spatial image feature. We encode the question Q with a 2-layer $LSTM$, which we refer to as $f^Q(Q)$. We combine this and the spatial image feature using element-wise multiplication followed by signed square-root, L2 normalization, and Dropout, and two more layers of 1×1 convolutions with ReLU in between. This process gives us a $N \times M$ attention map $\bar{\alpha}_{n,m}$. We apply softmax to produce a normalized soft attention map.

The attention map is then used to take the weighted sum over the image features and this representation is once again combined with the LSTM feature to predict the answer $\hat{y}$ as a classification problem over all answers Y. We provide an extended formalized version in the supplemental.

Multimodal explanation model. We argue that to generate multimodal explanation, we should condition the explanation on question, answer, and image. We model this by pooling the image, question, and answer representations to generate an attention map, our *Visual Pointing*. The *Visual Pointing* is further used to create attention features that guide the generation of our *Textual Justification*.

More specifically, the answer predictions are embedded in a d-dimensional space followed by tanh non-linearity and a fully connected layer: $f^{yEmbed}(\hat{y}) = W_6(tanh(W_5\hat{y} + b_5)) + b_6$. To allow the model to learn how to attend to relevant spatial location based on the answer, image, and question, we combine this answer feature with Question-Image embedding $\bar{f}^{IQ}(I, Q)$ from the answering model. Applying 1×1 convolutions, element-wise multiplication followed by signed square-root, L2 normalization, and Dropout, results in a multimodal feature.

$$\begin{aligned} \bar{f}^{IQA}(I, n, m, Q, \hat{y}) =& (W_7 \bar{f}^{IQ}(I, Q, n, m) + b_7) && (2.1) \\ & \odot f^{yEmbed}(\hat{y})) && (2.2) \\ f^{IQA}(I, Q, \hat{y}) =& L2(signed\_sqrt(\bar{f}^{IQA}(I, Q, \hat{y}))) && (2.3) \end{aligned}$$

Next we predict a $N \times M$ attention map $\bar{\alpha}_{n,m}$ and apply softmax to produce a normalized soft attention map, our *Visual Pointing* $\alpha^{pointX}_{n,m}$, which aims to point at the evidence of the generated explanation:

$$\begin{aligned} \bar{\alpha}_{n,m} =& f^{pointX}(I, n, m, Q, \hat{y}) && (2.4) \\ =& W_9 \rho(W_8 f^{IQA}(I, Q, \hat{y}) + b_8) + b_9 && (2.5) \\ \alpha^{pointX}_{n,m} =& \frac{\exp(\bar{\alpha}_{n,m})}{\sum_{i=1}^{N} \sum_{j=1}^{M} \exp(\bar{\alpha}_{n,m})} && (2.6) \end{aligned}$$

with Relu $\rho(x) = max(x, 0)$.

Using $\alpha^{pointX}_{n,m}$, we compute the attended visual representation, and merge it with the LSTM feature that encodes the question and the embedding feature that encodes the answer:

$$\begin{aligned} f^{X}(I, Q, \hat{y}) =& (W_{10} \sum_{x=1}^{N} \sum_{y=1}^{M} \alpha^{pointX}_{n,m} f^{I}(I, n, m) + b_{10}) && (2.7) \\ & \odot (W_{11} f^{Q}(Q) + b_{11}) \odot f^{yEmbed}(\hat{y}) && (2.8) \end{aligned}$$

This combined feature is then fed into an LSTM decoder to generate our Textual Justifications that are conditioned on image, question, and answer.

Textual Justifications are a sequence of words $[\boldsymbol{w_1}, \boldsymbol{w_2}, \boldsymbol{\ldots}]$ and our model predicts one word w_t at each time step t conditioned on the previous word and the hidden state of the LSTM:

$$\begin{aligned} h_t = & f^{LSTM}(f^{X}(I, Q, \hat{y}), w_{t-1}, h_{t-1}) && (2.9) \\ w_t = & f^{pred}(h_t) = Softmax(W_{pred} h_t + b_{pred}) && (2.10) \end{aligned}$$

2.5 Experiments

In this section, we present quantitative results on ablations done for textual justification and visual pointing tasks, and discuss their implications. Additionally, we provide and analyze qualitative results for both tasks.

	GT-ans	Train	Att.			VQA-X						ACT-X			
	Cond.	Data	Expl.	B	M	R	C	S	Human	B	M	R	C	S	Human
[43]	Yes	Desc.	No	–	–	–	–	–	–	12.9	15.9	39.0	12.4	12.0	17.4
Ours on Descriptions	Yes	Desc.	Yes	6.1	12.8	26.4	36.2	12.1	34.5	6.9	12.9	28.3	20.3	7.3	22.9
Ours w/o Attention	Yes	Expl.	No	18.0	17.6	42.4	66.3	14.3	40.1	16.9	17.0	42.0	33.3	10.6	21.4
Ours	Yes	Expl.	Yes	**19.8**	**18.6**	**44.0**	**73.4**	**15.4**	**45.1**	**24.5**	**21.5**	**46.9**	**58.7**	**16.0**	**38.2**
Ours on Descriptions	No	Desc.	Yes	5.9	12.6	26.3	35.2	11.9	–	5.2	11.0	26.5	10.4	4.6	–
Ours w/o Attention	No	Expl.	No	18.0	17.3	42.1	63.6	13.8	–	11.9	13.6	37.9	16.9	5.7	–
Ours	No	Expl.	Yes	**19.5**	**18.2**	**43.4**	**71.3**	**15.1**	–	**15.3**	**15.6**	**40.0**	**22.0**	**7.2**	–

Table 2.2: Evaluation of Textual Justifications. BLEU-4 (B), METEOR (M), ROUGE (R), CIDEr (C), and SPICE (S). All in %. GT-ans Cond. stands for Ground Truth Answers Conditioning, Train Data represents the type of data used for training, Att. Expl. denotes whether attention mechanism is used when generating explanations, and Human indicates human evaluation scores.

2.5.1 Experimental Setup

Here, we detail our experimental setup in terms of model training, hyperparameter settings, and evaluation metrics.

Model training and hyperparameters. For VQA, the answering model of PJ-X is pre-trained on the VQA v2 training set [59]. We then freeze or finetune the weights of the answering model when training the multimodal explanation model on textual annotations as VQA-X is significantly smaller than the original VQA dataset. For activity recognition, answering and explanation components of PJ-X are trained jointly. The spatial feature size of PJ-X is $N = M = 14$. For VQA, the answer space is limited to the 3000 most frequent answers on the training set (*i.e.* $|Y| = 3000$) whereas for activity recognition, $|Y| = 397$. The answer embedding size is $d = 300$ for both tasks.

Evaluation metrics. We evaluate our textual justifications w.r.t BLEU-4 [62], METEOR [63], ROUGE [64], CIDEr [65] and SPICE [66] metrics, which measure the degree of similarity between generated and ground truth sentences. We also include human evaluation since automatic metrics do not always reflect human preference. We randomly choose 1000 data points each from the test splits of VQA-X and ACT-X datasets, where the model predicts the correct answer, and then for each data point ask 3 human subjects to judge whether a generated explanation is better than, worse than, or equivalent to the ground truth explanation (we note that human judges do not know what explanation is ground truth and the order of sentences is randomized). We report the percentage of generated explanations which are equivalent to or better than ground truth human explanations, when at least 2 out of 3 human judges agree.

For visual pointing task, we use Earth Mover's Distance (EMD) [67] which measures the distance between two probability distributions over a region. To compute EMD, we use [68]. We also report on Rank Correlation which was used in [44]. For computing Rank Correlation, we follow [44] where we scale the generated attention map and the human ground-truth annotations from the VQA-X/ACT-X/VQA-HAT datasets to 14×14, rank the pixel values, and then compute correlation between these two ranked lists.

2.5.2 Textual Justification

We ablate PJ-X and compare with related approaches on our VQA-X and ACT-X datasets through automatic and human evaluations for the generated explanations.

Details on compared models. We compare with the state-of-the-art [43] using publicly available code and use ResNet features for fair comparison. The generated sentences from [43] are conditioned on both the image and the class label and uses a discriminative loss. The discriminative loss requires training a sentence classifier and back-propagating policy gradients when training the language generator. Our model does not use discriminative loss/policy gradients and does not require defining a reward. Note that [43] is trained with descriptions. Similarly, "Ours on Descriptions" is an ablation in which we train PJ-X on descriptions instead of explanations. "Ours w/o Attention" is similar to [43] in the sense that there is no attention mechanism involved when generating explanations, however, it does not use the discriminative loss and is trained on explanations instead of descriptions. For all models, explanations can be generated either by conditioning on ground-truth labels or on predicted labels. We call the former "GT-ans Conditioning" and show results in Table 2.2 to see how it affects the performance.

Descriptions vs. Explanations. "Ours" significantly outperforms "Ours with Descriptions" by a large margin on both datasets which is expected as descriptions are insufficient for the task of generating explanations. Additionally, "Ours" compares favorably to [43] even in the case when "Ours" generates textual justifications conditioned on the prediction, not the ground-truth answer. These results demonstrate the limitation of training explanation systems with descriptions, and thus support the necessity of having datasets specifically curated for explanations. "Ours on Descriptions" performs worse on certain metrics compared to [43] which may be attributed to additional training signals generated from discriminative loss and policy gradients, but further investigation is left for future work.

Unimodal explanations vs. Multimodal explanations. Including attention when generating textual justifications allows us to build a multimodal explanation model. Aside from the immediate benefit of providing visual rationale about a decision,

	Earth Mover's		Rank Correlation		
	(lower is better)		(higher is better)		
	VQA-X	ACT-X	VQA-X	ACT-X	VQA-HAT
Random Point	6.71	6.59	+0.0017	+0.0003	-0.0001
Uniform	3.60	3.25	+0.0003	-0.0001	-0.0007
HieCoAtt-Q [44]	–	–	–	–	+0.2640
Answering Model	2.77	4.78	+0.2211	+0.0104	+0.2234
Ours	**2.64**	**2.54**	**+0.3423**	**+0.3933**	**+0.3964**

Table 2.3: Evaluation of Visual Pointing Justifications. For rank correlation, all results have standard error < 0.005.

learning to point at visual evidence helps generate better textual justifications. As can be seen in Table 2.2, "Ours" greatly improves textual justifications compared to "Ours w/o Attention" on both datasets, demonstrating the value of multimodal explanation systems.

2.5.3 Visual Pointing

We compare the visual pointing performance of PJ-X to several baselines and report quantitative results.

Details on compared models. We compare our model against the following baselines. *Random Point* randomly attends to a single point in a 14×14 grid. *Uniform Map* generates attention map that is uniformly distributed over the 14×14 grid. We also compare PJ-X attention maps with those generated from state-of-the-art VQA systems ([44]).

Improved localization with textual explanations. We evaluate attention maps using the Earth Mover's Distance (lower is better) and Rank Correlation (higher is better) on VQA-X and ACT-X in Table 2.3. From Table 2.3, we observe that "Ours" outperforms baselines *Random Point* and *Uniform Map*, as well as our answering model and [44] on both datasets and on both metrics. The attention maps generated from our answering model and [44] do not receive training signals from the textual annotations as they are only trained to predict the correct answer, whereas the attention maps generated from PJ-X multimodal explanation model are latently learned through supervision of textual annotations. This implies that learning to generate textual explanations helps improve visual pointing task, and further confirms the advantages of multimodal explanations.

2.5.4 Qualitative Results

In this section we present our qualitative results on VQA-X and ACT-X datasets demonstrating that our model generates high quality sentences and the attention maps point to relevant locations in the image.

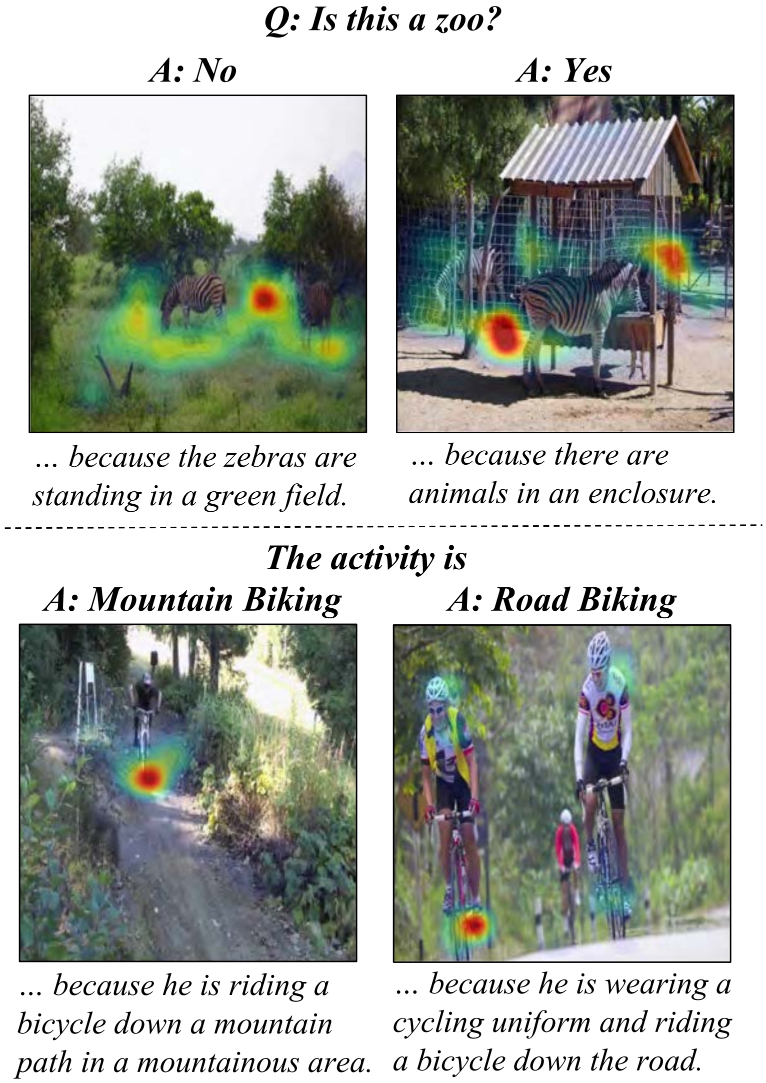

Q: Is this a zoo?
A: No
A: Yes
... because the zebras are standing in a green field.
... because there are animals in an enclosure.
The activity is
A: Mountain Biking
A: Road Biking
... because he is riding a bicycle down a mountain path in a mountainous area.
... because he is wearing a cycling uniform and riding a bicycle down the road.

Figure 2.5: Qualitative results on VQA-X (top row) and ACT-X (bottom row): For each image the PJ-X model provides an answer and a justification, and points to the evidence for that justification. For VQA-X, we show complementary pairs. Images are from [1] and [2].

VQA-X. As seen in Figure 2.5, our textual justifications are able to both capture common sense and discuss specific image parts important for answering a question. For example, when asked “Is this a zoo?”, the explanation model is able to discuss what the concept of “zoo” represents (*i.e.* “animals in an enclosure”) and also discuss specific regions (*i.e.* “green field”) to determine whether it is a zoo or not.

Visually, we notice that our attention model is able to point to important visual evidence as well. For example in the top row of Figure 2.5, the visual explanation focuses on the field in one case, and the fence in another.

ACT-X. Figure 2.5 also shows results on our ACT-X dataset. Textual explanations discuss a variety of visual cues important for correctly classifying activities such as global context (*e.g.* “a mountainous area”), and person-object interaction, (*e.g.* “riding a bicycle”) for mountain biking. These explanations require determining which of many multiple cues are appropriate to justify a particular action.

Our model points to visual evidence important for understanding each human activity. For example to classify “mountain biking” in the bottom row of Figure 2.5 the model focuses both on the bicycle as well as the mountainous path. Our model can also differentiate between similar activities based on the context, *e.g.*”mountain biking”/”road biking”.

Explanation Consistent with Incorrect Prediction. Generating reasonable explanations for correct answers is important, but it is also crucial to see how a system behaves when predictions are incorrect. Such analysis would provide insights into

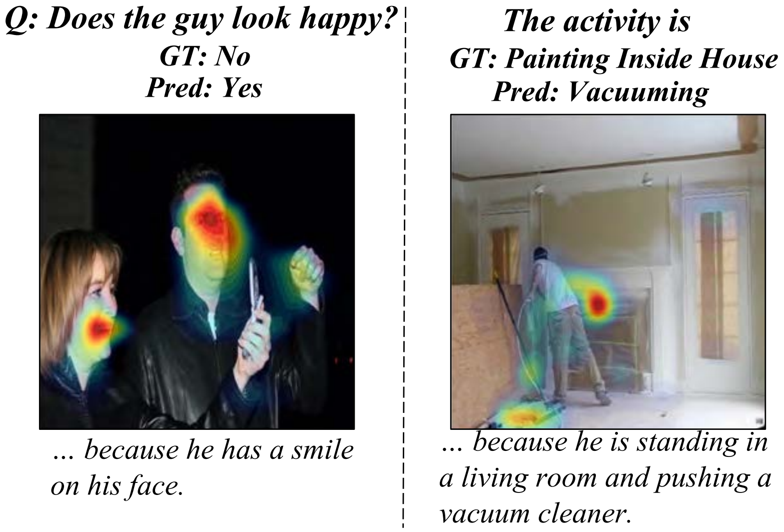

Q: Does the guy look happy?
GT: No
Pred: Yes
... because he has a smile on his face.
The activity is
GT: Painting Inside House
Pred: Vacuuming
... because he is standing in a living room and pushing a vacuum cleaner.

Figure 2.6: Visual and textual explanations generated by our model conditioned on incorrect predictions. Images are from [1] and [2].

whether the explanation generation component of the model is consistent with the answer prediction component. In Figure 2.6, we can see that the explanations are consistent with the incorrectly predicted answer for both VQA-X and ACT-X. For instance in the right example, we see that the model attends to a vacuum-like object and textually justifies the prediction "vacuuming". Such consistency between the answering model and the explanation model is also shown in Table 2.2 where we see a drop in performance when explanations are conditioned on predictions (bottom rows) instead of the ground-truth answers (top rows).

2.5.5 Usefulness of Multimodal Explanations

In this section, we address some of the advantages of generating multimodal explanations. In particular, we look at cases where visual explanations are more informative than textual explanations, and vice versa. We also investigate how multimodal explanations can help humans diagnose the performance of an AI system.

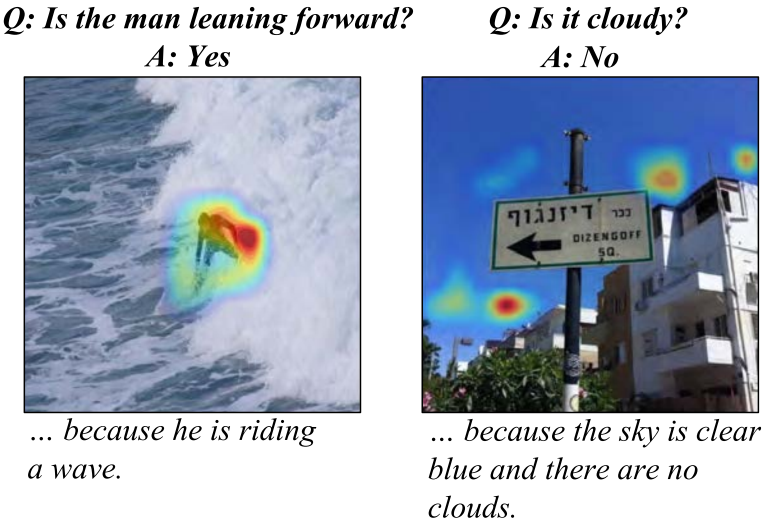

Q: Is the man leaning forward?
A: Yes
... because he is riding a wave.
Q: Is it cloudy?
A: No
ככר דיזנגוף
DIZENGOFF SQ.
... because the sky is clear blue and there are no clouds.

Figure 2.7: Qualitative results comparing the insightfulness of visual pointing and textual justification. The left example demonstrates how visual pointing is more informative than textual justification whereas the right example shows the opposite. Images are from [1].

Complementary Explanations. Multimodal explanations can support different tasks or support each other. Interestingly, in Figure 2.7, we present some examples where visual pointing is more insightful than textual justification, and vice versa. Looking at the left example in Figure 2.7, it is rather difficult to explain "leaning" with language and the model resorts to generating a correct, yet uninsightful sentence.

	VQA-X	ACT-X
Without explanation	57.5%	51.5%
Ours on Descriptions	66.5%	72.5%
Ours w/o Attention	61.5%	76.5%
Ours	**70.0%**	**80.5%**

Table 2.4: Accuracy of humans guessing whether the model correctly or incorrectly answered the question.

However, the concept is easily conveyed when looking at the visual pointing result. In contrast, the right example shows the opposite. Looking at only some patches of the sky presented by the visual pointing result does not necessarily confirm if the scene is cloudy or not, while it is also unclear if attending to the entire region of the sky is a desired behavior. Yet, the textual justification succinctly captures the rationale. These examples clearly demonstrate the value of generating multimodal explanations.

Diagnostic Explanations. We evaluate an auxiliary task where humans have to guess whether the system correctly or incorrectly answered the question. The predicted answer is not shown; only image, question, correct answer, and textual/visual explanations. The set contains 50% correctly answered questions. We compare our model against the models used for ablations in Table 2.2. Table 2.4 indicates that explanations are better than no explanations and our model is more helpful than models trained on descriptions and also models trained to generate textual explanations only.

2.6 Conclusion

As a step towards explainable AI models, we proposed multimodal explanations for real-world tasks. Our model is the first to be capable of providing natural language justifications of decisions as well as pointing to the evidence in an image. We have collected two novel explanation datasets through crowd sourcing for visual question answering and activity recognition, *i.e.* VQA-X and ACT-X. We quantitatively demonstrated that learning to point helps achieve high quality textual explanations. We also quantitatively show that using reference textual explanations to train our model helps achieve better visual pointing. Furthermore, we qualitatively demonstrated that our model is able to point to the evidence as well as to give natural sentence justifications, similar to ones humans give. Our model is a third-person, post-hoc rationalization type of explanation, akin to what one human produces when asked to explain the actions of a second human. A third-person explanation is clearly different from a first-person explanation, but we believe both forms of explanation are valuable.

Chapter 3

Discovering Non-monotonic Autoregressive Orderings with Variational Inference

3.1 Introduction

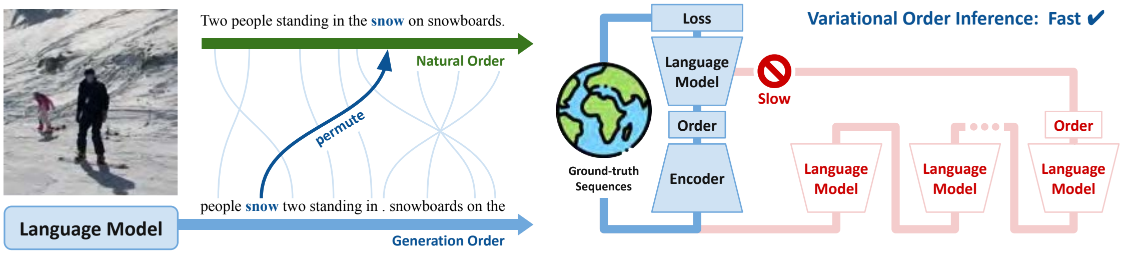


Figure 3.1: Left: our language model, shown in light blue, learns to decode in non-monotonic generation orders, rather than pre-determined orders, such as left-to-right. Right: during training, we leverage an encoder in a variational inference pipeline to parameterize a latent distribution over the generation orders for the autoregressive language model. In this way, training can be done in just one forward / backward pass per batch, unlike previous approaches in non-monotonic sequence modeling that require multiple forward passes per batch to determine a generation order.

Autoregressive models have a rich history. Early papers that studied autoregressive models, such as [69] and [70], showed an interest in designing algorithms that did not require a gold-standard autoregressive order to be known upfront by researchers. However, these papers were overshadowed by developments in natural

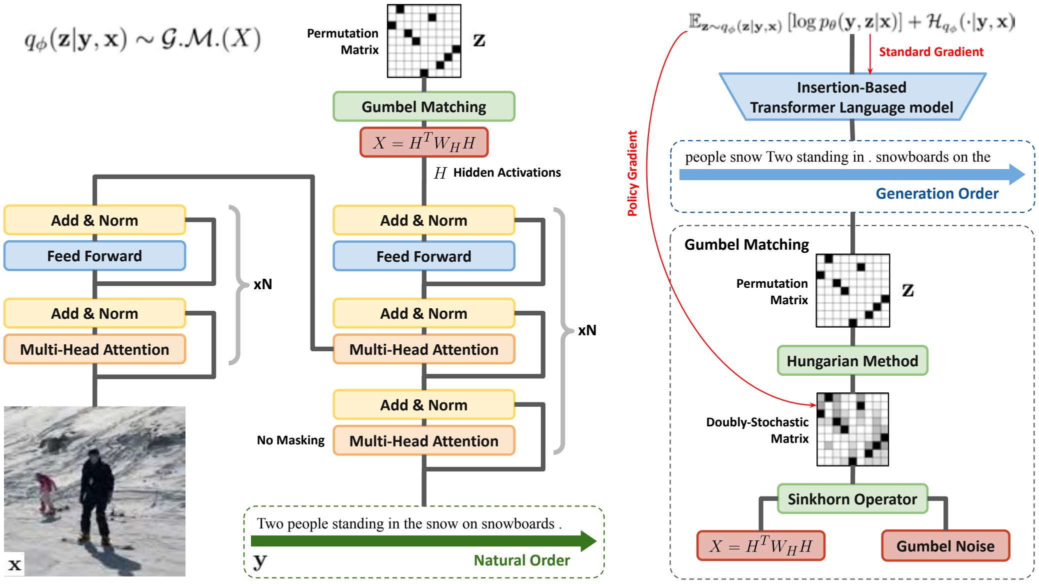


Figure 3.2: Architecture for sequence-modeling tasks. The goal is to predict the target sequence $\mathbf{y}$ given the source sequence $\mathbf{x}$, with latent generation orders $\mathbf{z}$ represented as permutation matrices. We use a Transformer without causal masking to serve as the encoder in *Variational Order Inference* (VOI), which samples orderings in a single forward pass. These orderings are used to train an insertion-based Transformer language model, which serves as the VOI decoder. As the objective is non-differentiable over permutation matrices, policy gradient algorithms (e.g. Reinforce [3], PPO [4]) are applied to update the permutation-generating encoder.

language processing that demonstrated the power of the left-to-right autoregressive order [71, 72]. Since then, the left-to-right autoregressive order has been essential for application domains such as image captioning [73, 74], machine translation [75, 76] and distant fields like image synthesis [77]. However, interest in non left-to-right autoregressive orders is resurfacing [78, 79], and evidence [80–82] suggests adaptive orders may produce more accurate autoregressive models. These positive results make designing algorithms that can leverage adaptive orders an important research domain.

Inferring autoregressive orderings in a data-driven manner is challenging. Modern benchmarks for machine translation [83] and other tasks [84] are not labelled with gold-standard orders, and left-to-right seems to be the default. This could be explained if domain-independent methodology for identifying *high-quality* orders is an open question. Certain approaches [78, 79, 85] use hand-designed loss functions to promote a *genre* of orders—such as balanced binary trees. These loss functions incorporate certain domain-assumptions: for example, they assume the balanced binary tree order will not disrupt learning. Learning disruption is an important consideration, because prior work shows that poor orders may prohibitively slow learning [86]. Future approaches to inferring autoregressive orders should withhold domain knowledge, to promote their generalization.

To our best knowledge, we propose the first domain-independent unsupervised learner that discovers high-quality autoregressive orders through fully-parallelizable end-to-end training without domain-specific tuning. We provide three main contributions that stabilize this learner. First, we propose an encoder architecture that conditions on training examples to output autoregressive orders represented as permutation matrices using techniques in combinatorial optimization. Second, we propose *Variational Order Inference* (VOI) that learns an approximate posterior over autoregressive orders. Finally, we develop a practical algorithm for solving the resulting non-differentiable ELBO end-to-end with policy gradients. A high-level summary of our approach is presented in Figure 3.1, and a detailed architecture diagram for sequence modeling tasks is presented in Figure 3.2.

Empirical results with our solution on various sequence modeling tasks suggest that with similar hyperparameters, our algorithm is capable of recovering autoregressive orders that are even better than fixed orders. Case studies suggest that our learned orders depend adaptively on content, and resemble a type of *best-first* generation order, which prioritizes salient objects / phrases and deprioritizes auxillary tokens (see Fig. 3.3).

3.2 Related Works

Autoregressive Models Autoregressive models decompose the generation of a high dimensional probability distribution by generating one dimension at a time, with a predefined order. Combined with high capacity neural networks, this approach to modeling complex distributions has been very successful [87, 88]. Recent works have achieved great improvements with autoregressive models in many applications, including language modeling [13, 89, 90], machine translation [12] and image captioning [91]. Most previous works on autoregressive models regress to an ordering selected by designers, with left-to-right emerging as the primary choice. In contrast, our method is capable of learning arbitrary orderings conditioned on data and is more flexible.

Non-Monotonic Autoregressive Orderings [92] shows that a sub-optimal ordering can severely limit the viability of a language model and propose to first generate a partially filled sentence template and then fill in missing tokens. Previous works have

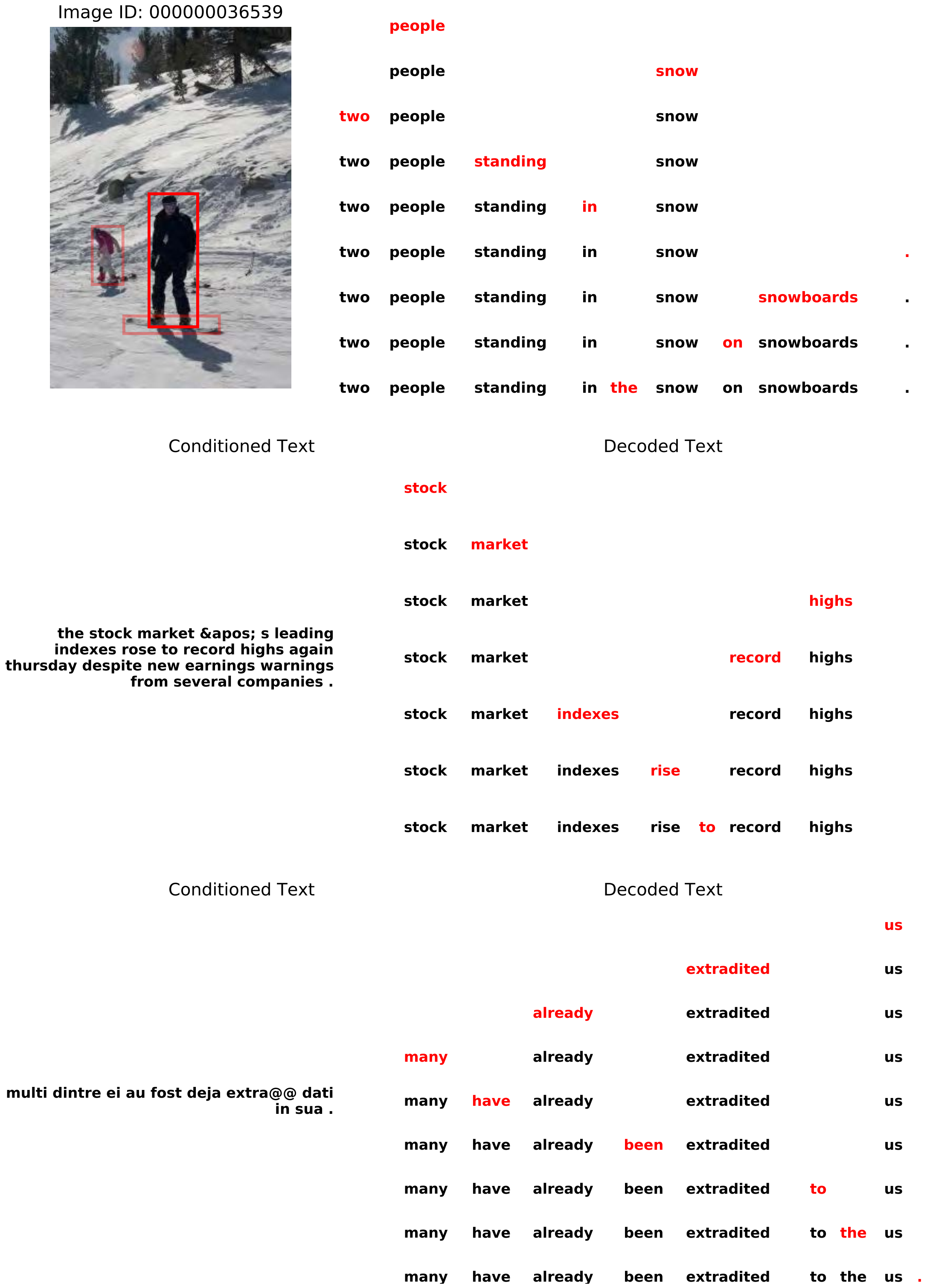


Image ID: 000000036539
people
people snow
two people snow
two people standing snow
two people standing in snow
two people standing in snow .
two people standing in snow snowboards .
two people standing in snow on snowboards .
two people standing in the snow on snowboards .
Conditioned Text
Decoded Text
the stock market &apos; s leading indexes rose to record highs again thursday despite new earnings warnings from several companies .
stock
stock market
stock market highs
stock market record highs
stock market indexes record highs
stock market indexes rise record highs
stock market indexes rise to record highs
Conditioned Text
Decoded Text
multi dintre ei au fost deja extra@@ dati in sua .
us
extradited us
already extradited us
many already extradited us
many have already extradited us
many have already been extradited us
many have already been extradited to us
many have already been extradited to the us
many have already been extradited to the us .

Figure 3.3: Examples of sequence generations for tasks like image captioning, text summarization, and machine translation using the decoder insertion-based language model (top right of Fig. 3.2) in *Variational Order Inference* . Orderings highlight descriptive phrases from conditioned images (e.g., “people”, “snow”) and sentences (e.g., “stock market”, “U.S.”), while modifiers (e.g., “to”, “on”, “the”) come last.

also studied bidirectional decoding [93–95] and syntax trees based decoding [96–100] in the natural language setting. However, all of the works mentioned above do not learn the orderings and instead opt to use heuristics to define them. [101] performs language modeling according to a known prior, such as balanced binary tree, and does not allow arbitrary sequence generation orders. [102] proposes to use a tree-based recursive generation method to learn arbitrary generation orders. However, their performance lags behind that of left-to-right. [103] proposes Transformer-InDIGO to allow non-monotonic sequence generation by first pretraining with pre-defined orderings, such as left-to-right, then fine-tuning use Searched Adaptive Order (SAO) to find alternative orderings. They report that without pretraining, the learned orders degenerate. In addition, they perform beam search to acquire plausible orderings, which cannot be efficiently parallelized across different time-steps. [104] proposes an alternative to SAO, but suffers from similar poor time complexity. In contrast, our method learns high-quality orderings directly from data under fully-parallelizable end-to-end training.

Variational Methods Our method optimizes the evidence lower bound, or ELBO in short. ELBO is a quantity that is widely used as an optimization proxy in the machine learning literature, where the exact quantity is hard to compute or optimize. Variational methods have achieved great success in machine learning, such as VAE [105] and β-VAE [106].

Combinatorial Optimization Recent works have studied gradient-based optimization in the combinatorial space of permutations [107–109]. These works have been applied in tasks such as number sorting, jigsaw puzzle solving, and neural signal identification in worms. To our best knowledge, we are the first to build on these techniques to automatically discover autoregressive orderings in vision and language datasets.

3.3 Preliminaries

The goal of autoregressive sequence modelling is to model an ordered sequence of target values $\mathbf{y} = (y_1, y_2 \ldots, y_n) : y_i \in \mathbb{R}$, possibly conditioned on an ordered sequence of source values $\mathbf{x} = (x_1, x_2 \ldots, x_m) : x_i \in \mathbb{R}$, where $(\mathbf{x}, \mathbf{y})$ is sampled from the dataset $\mathcal{D}$. In the context of language modeling, $x_i, y_i \in \mathbb{N}$ as the token distribution is categorical.

Inspired by [110] and [103], we formulate the generation process of $\mathbf{y}$ as a $2n$ step process, where at time step $2t-1$ we generate a value, and at timestep $2t$ we select a not-yet-chosen position in $\{1, 2, \cdots, n\}$ to insert the value. Thus, we introduce the latent sequence variable $\mathbf{z} = (z_1, z_2 \ldots, z_n) : \mathbf{z} \in S_n$, where S_n is the set

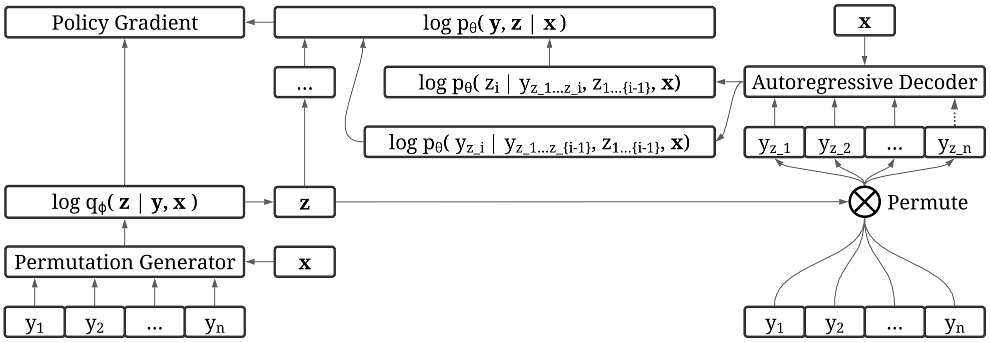

Policy Gradient
log p_θ($\mathbf{y}, \mathbf{z}$ | $\mathbf{x}$)
$\mathbf{x}$
...
log p_θ(z_i | $y_{z\_1...z\_i}$, $z_{1...\{i-1\}}$, $\mathbf{x}$)
Autoregressive Decoder
log p_θ(y_{z_i} | $y_{z\_1...z\_\{i-1\}}$, $z_{1...\{i-1\}}$, $\mathbf{x}$)
y_{z_1}
y_{z_2}
...
y_{z_n}
log q_ϕ($\mathbf{z}$ | $\mathbf{y}, \mathbf{x}$)
$\mathbf{z}$
Permute
Permutation Generator
$\mathbf{x}$
y_1
y_2
...
y_n
y_1
y_2
...
y_n

Figure 3.4: Computational diagram for the encoder (left) and decoder (right) that compose Variational Order Inference. We optimize a lower bound on the standard maximum likelihood objective.

of one-dimensional permutations of $\{1, 2, \cdots, n\}$, and z_t is defined as the absolute position of the value generated at time step $2t - 1$ in the naturally ordered $\mathbf{y}$. Then $p(\mathbf{y}, \mathbf{z}|\mathbf{x})$ denotes the probability of generating $\mathbf{y}$ in the ordering of $\mathbf{z}$ given the source sequence $\mathbf{x}$. We can thus factorize $p(\mathbf{y}, \mathbf{z}|\mathbf{x})$ using the chain rule:

$$p(\mathbf{y}, \mathbf{z}|\mathbf{x}) = p(y_{z_1}|\mathbf{x})p(z_1|y_{z_1}, \mathbf{x}) \prod_{i=2}^{n} p(y_{z_i}|z_{<i}, y_{z_{<i}}, \mathbf{x})p(z_i|z_{<i}, y_{z_{<=i}}, \mathbf{x}) \tag{3.1}$$

For example, $p(y_1, y_2, z_1 = 2, z_2 = 1|\mathbf{x}) = p(y_2|\mathbf{x})p(z_1|y_2, \mathbf{x})p(y_1|z_1, y_2, \mathbf{x})p(z_2|y_1, z_1, y_2, \mathbf{x})$ is defined as the probability of generating y_2 in the first step, then inserting y_2 into absolute position 2, then generating y_1, and finally inserting y_1 into absolute position 1.

Note that in practice, the length of $\mathbf{y}$ is usually varied. Therefore, we do not first create a fixed-length sequence of blanks and then replace the blanks with actual values. Instead, we dynamically insert a new value at a position relative to the previous values. One common approach to predict such relative position is Pointer Network [110]. In other words, at timestep t, we insert the value at position r_t relative to the previous generated values. Here, for any $\mathbf{z} \in S_n$, $\mathbf{r} = (r_1, r_2, \ldots, r_n)$ is constructed such that there is a bijection between S_n and the set of all constructed $\mathbf{r}$. Due to such bijection, we can use $\mathbf{z}$ and $\mathbf{r}$ interchangeably. We will use $\mathbf{z}$ throughout the paper.

3.4 Variational Order Inference (VOI)

Starting from just the original data $\mathbf{y}$ in natural order, we can use variational inference to create an objective (3.2) that allows us to recover latent order $\mathbf{z}$, parametrized by two neural networks θ and ϕ. The encoder network ϕ samples autoregressive orders given the ground truth data, which the decoder network θ uses to recover $\mathbf{y}$. More specifically, ϕ is a non-autoregressive network (permutation generator in Fig. 3.4) that takes in the source sequence $\mathbf{x}$ and the entire ground truth target sequence $\mathbf{y}$ and outputs latent order $\mathbf{z}$ in a single forward pass. θ is an autoregressive network (autoregressive decoder in Fig. 3.4) that takes in $\mathbf{x}$ and predicts both the target sequence $\mathbf{y}$ and the ordering $\mathbf{z}$ through the factorization in Equation (3.1). We name this process *Variational Order Inference* (VOI).

$$\begin{aligned} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}\left[\log p_\theta(\mathbf{y}|\mathbf{x})\right] &= \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}\left[\log \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{y},\mathbf{x})}\left[\frac{p_\theta(\mathbf{y},\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{y},\mathbf{x})}\right]\right] \\ &\geq \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}\left[\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{y},\mathbf{x})}\left[\log p_\theta(\mathbf{y},\mathbf{z}|\mathbf{x})\right] + \mathcal{H}_{q_\phi}(\cdot|\mathbf{y},\mathbf{x})\right] \end{aligned} \tag{3.2}$$

Here, $\mathcal{H}_{q_\phi}$ is the entropy term. In practice, a closed form for $\mathcal{H}_{q_\phi}$ usually cannot be obtained, so an approximation is needed. During training, we train ϕ and θ jointly to maximize the ELBO in (3.2). During testing, we only keep the decoder θ.

To optimize the decoder network θ in (3.2), for each $\mathbf{y}$, we first sample K latents $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_K$ from $q_\phi(\cdot|\mathbf{y},\mathbf{x})$. We then update θ using the Monte-Carlo gradient estimate $\mathbb{E}_{\mathbf{y}\sim\mathcal{D}}\left[\frac{1}{K}\sum_{i=1}^{K}\nabla_\theta \log p_\theta(\mathbf{y},\mathbf{z}_i|\mathbf{x})\right]$.

Optimizing the encoder network ϕ is tricky. Since $\mathbf{z}$ is a discrete latent variable, the gradient from $\log p_\theta(\mathbf{y},\mathbf{z})$ does not flow through $\mathbf{z}$. Thus, we formulate (3.2) in a reinforcement learning setting with a one-step Markov Decision Process $(\mathcal{S},\mathcal{A},\mathcal{R})$. Under our setting, the state space $\mathcal{S}=\mathcal{D}$; for each state $(\mathbf{x},\mathbf{y})\in\mathcal{D}$, the action space $\mathcal{A}_{(\mathbf{x},\mathbf{y})} = S_{\text{length}(\mathbf{y})}$ with entropy term $\mathcal{H}_{q_\phi}(\cdot|\mathbf{y},\mathbf{x})$; the reward function $\mathcal{R}((\mathbf{x},\mathbf{y}),\mathbf{z}\in S_{\text{length}(\mathbf{y})}) = \log p_\theta(\mathbf{y},\mathbf{z}|\mathbf{x})$. We can then set the optimization objective $L(\phi)$ to be (3.2). In practice, we find that adding an entropy coefficient β and gradually annealing it can speed up the convergence of decoder while still obtaining good autoregressive orders.

To compute $\nabla_\phi L(\phi)$, we derive the policy gradient with baseline formulation [3]:

$$\nabla_\phi L(\phi) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}\left[\mathbb{E}_{\mathbf{z}\sim q_\phi}\left[\nabla_\phi \log q_\phi(\mathbf{z}|\mathbf{y},\mathbf{x})(\log p_\theta(\mathbf{y},\mathbf{z}|\mathbf{x}) - b(\mathbf{y},\mathbf{x}))\right] + \beta\nabla_\phi\mathcal{H}_{q_\phi}\right] \tag{3.3}$$

where $b(\mathbf{y},\mathbf{x})$ is the baseline function independent of action $\mathbf{z}$. The reason we use a state-dependent baseline $b(\mathbf{y},\mathbf{x})$ instead of a global baseline b is that the the length of $\mathbf{y}$ can have a wide range, causing significant reward scale difference. In particular, we set $b(\mathbf{y},\mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim q_\phi}\left[\log p_\theta(\mathbf{y},\mathbf{z}_i|\mathbf{x})\right]$. If we sample $K \geq 2$ latents for each $\mathbf{y}$, then we can use its Monte-Carlo estimate $\frac{1}{K}\sum_{i=1}^{K}\log p_\theta(\mathbf{y},\mathbf{z}_i|\mathbf{x})$.

Since we use policy gradient to optimize ϕ, we still need a closed form for the distribution $q_\phi(\mathbf{z}|\mathbf{y},\mathbf{x})$. Before we proceed, we define $\mathcal{P}_{n\times n}$ as the set of $n \times n$ permutation matrices, where exactly one entry in each row and column is 1 and all other entries are 0; $\mathcal{B}_{n\times n}$ as the set of $n \times n$ doubly stochastic matrices, i.e. non-negative matrices whose sum of entries in each row and in each column equals 1; $\mathbb{R}^+_{n\times n}$ as the set of non-negative $n \times n$ matrices. Note $\mathcal{P}_{n\times n} \subset \mathcal{B}_{n\times n} \subset \mathbb{R}^+_{n\times n}$.

To obtain $q_\phi(\mathbf{z}|\mathbf{y},\mathbf{x})$, we first write $\mathbf{z}$ in two-dimensional form. For each $\mathbf{z} \in S_n$, let $f_n(\mathbf{z}) \in \mathcal{P}_{n\times n}$ be constructed such that $f_n(\mathbf{z})_i = \text{one\_hot}(z_i)$, where $f_n(\mathbf{z})_i$ is the i-th row of $f_n(\mathbf{z})$. Thus f_n is a natural bijection from S_n to $\mathcal{P}_{n\times n}$, and we can rewrite q_ϕ as a distribution over $\mathcal{P}_{n\times n}$ such that $q_\phi(f_n(\mathbf{z})|\mathbf{y},\mathbf{x}) = q_\phi(\mathbf{z}|\mathbf{y},\mathbf{x})$.

Next, we need to model the distribution of $q_\phi(\cdot|\mathbf{y},\mathbf{x})$. Inspired by [107], we model $q_\phi(\cdot|\mathbf{y},\mathbf{x})$ as a Gumbel-Matching distribution $\mathcal{G}.\mathcal{M}.(X)$ over $\mathcal{P}_{n\times n}$, where $X = \phi(\mathbf{y},\mathbf{x}) \in \mathbb{R}^{n\times n}$ is the output of ϕ. Then for $P \in \mathcal{P}_{n\times n}$,

$$q_\phi(\mathbf{z}|\mathbf{y},\mathbf{x}) = q_\phi(f_n^{-1}(P)|\mathbf{y},\mathbf{x}) = q_\phi(P|\mathbf{y},\mathbf{x}) \propto \exp \langle X, P\rangle_F \tag{3.4}$$

where $\langle X, P\rangle_F = \text{trace}(X^T P)$ is the Frobenius inner product of X and P. To obtain samples in $\mathcal{P}_{n\times n}$ from the Gumbel-Matching distribution, [107] relaxes $\mathcal{P}_{n\times n}$ to $\mathcal{B}_{n\times n}$ by defining the Gumbel-Sinkhorn distribution $\mathcal{G}.\mathcal{S}.(X,\tau) : \tau > 0$ over $\mathcal{B}_{n\times n}$, and proves that $\mathcal{G}.\mathcal{S}.(X,\tau)$ converges almost surely to $\mathcal{G}.\mathcal{M}.(X)$ as $\tau \to 0^+$. Therefore, to approximately sample from $\mathcal{G}.\mathcal{M}.(X)$, we first sample from $\mathcal{G}.\mathcal{S}.(X,\tau)$, then apply Hungarian algorithm [111] to obtain $P \in \mathcal{G}.\mathcal{M}.(X)$. The entropy term H_{q_ϕ} can be approximated as $-\mathcal{D}_{KL}(\mathcal{G}.\mathcal{S}.(X,\tau) \mid\mid \mathcal{G}.\mathcal{S}.(\mathbf{0},\tau)) + \log n!$, and can be further approximated using the technique in Appendix B.3 of [107]. Further details are presented in Appendix.

The Gumbel-Matching distribution allows us to obtain the numerator for the closed form of $q_\phi(\mathbf{z}|\mathbf{y},\mathbf{x}) = q_\phi(f_n^{-1}(P)|\mathbf{y},\mathbf{x})$, which equals $\exp \langle X, P\rangle_F$. However, the denominator is intractable to compute and equals $\sum_{P\in\mathcal{P}_{n\times n}} \exp \langle X, P\rangle_F$. Upon further examination, we can express it as $\text{perm}(\exp(X))$, the matrix permanent of $\exp(X)$, and approximate it using $\text{perm}_B(\exp(X))$, its Bethe permanent. We present details about matrix permanent and Bethe permanent along with the proof that the denominator of $q_\phi(\cdot|\mathbf{y},\mathbf{x})$ equals $\text{perm}(\exp(X))$ in Appendix.

After we approximate q_ϕ, we can now optimize ϕ using the policy gradient in (3.3). We present a computational diagram of VOI in Figure 3.4, and a pseudocode of VOI in Algorithm 1. Note that even though latent space S_n is very large and contains $n!$ permutations, in practice, if $p_\theta(\mathbf{y},\mathbf{z}^*|\mathbf{x}) \geq p_\theta(\mathbf{y},\mathbf{z}|\mathbf{x})\ \forall \mathbf{z} \in S_n$, then $p_\theta(\mathbf{y},\mathbf{z}|\mathbf{x})$ tends to increase as the edit distance between $\mathbf{z}$ and $\mathbf{z}^*$ decreases. Therefore, ϕ does not need to search over the entire latent to obtain good permutations, making variational inference over S_n feasible.

Algorithm 1 Variational Order Inference

```
Given: encoder network $\phi$ with learning rate $\alpha_\phi$, decoder network $\theta$ with learning rate $\alpha_\theta$, entropy coefficient $\beta$, batch of training data $(\mathbf{X}, \mathbf{Y}) = \{(\mathbf{x}_b, \mathbf{y}_b)\}_{b=1}^{N}$ sampled from dataset $\mathcal{D}$
Set gradient accumulators $g_\phi = 0$, $g_\theta = 0$
for $(\mathbf{x}, \mathbf{y}) \in (\mathbf{X}, \mathbf{Y})$ do    ▷ In practice, this is done through parallel tensor operations
    $X = \phi(\mathbf{y}, \mathbf{x})$
    Sample $K$ doubly stochastic matrices $B_1, B_2, \ldots, B_K \in \mathcal{B}_{n\times n}$ from $\mathcal{G}.\mathcal{S}.(X, \tau)$
    Obtain $P_1, P_2, \ldots, P_K \in \mathcal{P}_{n\times n}$ from $B_1, B_2, \ldots, B_K$ using Hungarian Algorithm
    Obtain latents $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_K = f^{-1}_{\mathrm{len}(\mathbf{y})}(P_1), f^{-1}_{\mathrm{len}(\mathbf{y})}(P_2), \ldots, f^{-1}_{\mathrm{len}(\mathbf{y})}(P_K)$
    $g_\theta = g_\theta + \frac{1}{N\cdot K}\sum_{i=1}^{K} \nabla_\theta \log p_\theta(\mathbf{y}, \mathbf{z}_i|\mathbf{x})$
    Calculate $\log q_\phi(\mathbf{z}_i|\mathbf{y}, \mathbf{x}) = \langle X, P_i\rangle_F - \log(\mathrm{perm}(\exp(X)))$
                                  $\approx \langle X, P_i\rangle_F - \log(\mathrm{perm}_B(\exp(X)))$
    Calculate $b(\mathbf{y}, \mathbf{x}) = \frac{1}{K}\sum_{i=1}^{K} \log p_\theta(\mathbf{y}, \mathbf{z}_i|\mathbf{x})$
    $g_\phi = g_\phi + \frac{1}{N\cdot K}\sum_{i=1}^{K} \nabla_\phi \log q_\phi(\mathbf{z}_i|\mathbf{y}, \mathbf{x})(\log p_\theta(\mathbf{y}, \mathbf{z}_i|\mathbf{x}) - b(\mathbf{y}, \mathbf{x})) + \beta \cdot \nabla_\phi \mathcal{H}_{q_\phi}(\cdot|\mathbf{y}, \mathbf{x})$
end for
$\phi = \phi + \alpha_\phi \cdot g_\phi$
$\theta = \theta + \alpha_\theta \cdot g_\theta$
```

3.5 Experiments

Encoder and Decoder Architectures. We implement *Variational Order Inference* on conditional sequence generation tasks, specifically language modeling tasks. We implement the encoder of VOI as a Transformer with non-causal attention that outputs permutations in one forward pass. The generated permutations then serve as target generation orders for training an insertion-based Transformer language model. A summary of our architectures for conditional sequence generation tasks is illustrated in Figure 3.2. We would like to note that VOI is also applicable to unconditional sequence generation domains, such as image generation, through different encoder and decoder architectures, which we leave for future work. We would also like to note that “encoder" and “decoder" refer to the two networks ϕ and θ in Algorithm 1, respectively, instead of Transformer’s encoder and decoder.

For decoder θ, we use the Transformer-InDIGO [103] architecture, which maximizes $p_\theta(\mathbf{y}, \mathbf{z}|\mathbf{x})$ by alternating token generation and token insertion processes. Note that the ordering $\mathbf{z}$ used to train θ is obtained through the output of encoder ϕ in our approach, instead of through Searched Adaptive Order (SAO) proposed in the Transformer-InDIGO paper, which requires multiple forward passes per batch to obtain a generation order. Once $\mathbf{z}$ is already given, $p_\theta(\mathbf{y}, \mathbf{z}|\mathbf{x})$ can be optimized in one single pass through teacher forcing.

For encoder ϕ, we adopt the Transformer [112] architecture. Note that our encoder generates latents based on the entire ground truth target sequence $\mathbf{y}$. Therefore, it does not need to mask out subsequent positions during attention. We also experiment with different position embedding schemes (see Section 3.7) and find that Transformer-XL's [113] relative positional encoding performs the best, so we replace the sinusoid encoding in the original Transformer.

Tasks. We evaluate our approach on challenging sequence generation tasks: natural language to code generation (NL2Code) [114], image captioning, text summarization, and machine translation. For NL2Code, we use Django [84]. For image captioning, we use COCO 2017 [115]. For text summarization, we use English Gigaword [116, 117]. For machine translation, we use WMT16 Romanian-English (Ro-En).

Baselines. We compare our approach with several pre-defined fixed orders: Left-to-Right (L2R) [118], Common-First (Common) [119], Rare-First (Rare) [119], and Random-Ordering (Random). Here, Common-First order is defined as generating words with ordering determined by their relative frequency from high to low; Rare-First order is defined as the reverse of Common-First order; and Random-Ordering is defined as training with a randomly sampled order for each sample at each time step.

Preprocessing. For Django, we adopt the same preprocessing steps as described in [103], and we use all unique words as the vocabulary. For MS-COCO, we find that the baseline in [103] is much lower than commonly used in the vision and language community. Therefore, instead of using Resnet-18, we use the pretrained Faster-RCNN checkpoint using a ResNet-50 FPN backbone provided by TorchVision to extract 512-dimensional feature vectors for each object detection. To make our model spatially-aware, we also concatenate the bounding box coordinates for every detection before feeding into our Transformers' encoder. For Gigaword and WMT, we learn 32k byte-pair encoding (BPE, [120]) on tokenized data.

Training. For our decoder, we set $d_{\text{model}} = 512$, $d_{\text{hidden}} = 2048$, 6 layers for both Transformer's encoder and decoder, and 8 attention heads. This is the same model configuration as Transformer-Base [112] and as described in [103]. Our encoder also uses the same configuration. For our model trained with *Variational Order Inference* , we sample $K = 4$ latents for each training sample for Django, COCO, and Gigaword and $K = 3$ latents for WMT (due to computational resource constraints, we were unable to set a higher K for WMT). An ablation on the choices of K on a small dataset is presented in Section 3.7. For WMT, many previous works on nonsequential orderings [79] and nonautoregressive sequence generation [121] have found sequence-level knowledge distillation [122] helpful. Therefore, we first train the L2R model on the original WMT corpus, then create a new training corpus using beam search. We find that this improves the BLEU of VOI model by about 2.0. Even though the training set changed, the orderings learned by VOI are very similar to the ones trained on the original corpus. More detailed training processes are described in Appendix.

During training, our encoder and decoder are optimized in one single pass per batch. If we let N denote the batch size, l denote the length of each target sequence, and d denote the size of hidden vector, then one single forward pass of our model has computation complexity $O(NKdl^2)$, while Transformer-InDIGO trained with SAO has total complexity $O(Ndl^3)$. Since $K \ll l$ in general, our algorithm has better theoretical computational complexity during training. During evaluation, we only keep the decoder to iteratively generate the next position and token, which is as efficient as any standard fixed-order autoregressive models.

We also empirically compare VOI's runtime with that of SAO and fixed-order baselines (e.g. L2R). We implement SAO as described in [103]. We test the runtime on a single GPU in order to accurately measure the number of ops required. For training speed per iteration, we use a batch size of 8. For ordering search time, we use a batch size of 1 to avoid padding tokens in the input for accurate measure. We observe that VOI is significantly faster than SAO, which searches orderings sequentially. In practice, as we distribute VOI across more GPUs, the K factor in the runtime is effectively divided by the number of GPUs used (if we ignore the parallelization overhead), so we can achieve further speedups.

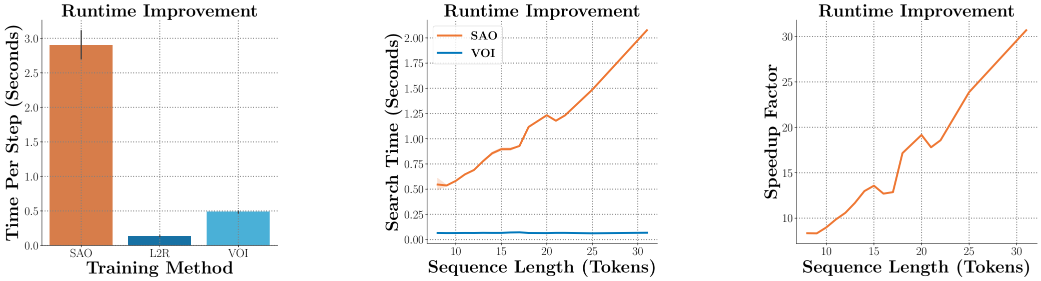


Figure 3.5: **Runtime performance improvement.** We compare the runtime performance of VOI ($K = 4$) with SAO on a single Tesla P100 GPU, in terms of time per training iteration and ordering search time. VOI outputs latent orderings in a single forward pass, and we observe a significant runtime improvement over SAO that searches orderings sequentially. The speedup factor linearly increases with respect to the sequence length.

Results. We compare VOI against predefined orderings along with Transformer-

InDIGO trained with SAO in Table 3.1. The metrics we used include BLEU-4 [123], Meteor [124], Rouge [125], CIDEr [126], and TER [127]. The "accuracy" reported for Django is defined as the percentage of perfect matches in code generation. Our results illustrate consistently better performance across fixed orderings. Most notably, CIDEr for MS-COCO, BLEU for Django, and Rouge-1 for Gigaword reveal the largest improvements in performance.

Order	MS-COCO				Django		Gigaword			WMT16 Ro-En		
	B	M	R-L	C	B	A	R-1	R-2	R-L	B↑	M↑	TER↓
InDIGO - SAO [1]	29.3	24.9	54.5	92.9	42.6	32.9	—	—	—	32.5	53.0	**49.0**
Ours - Random	28.9	24.2	55.2	92.8	21.6	26.9	30.1	11.6	27.6	20.3	43.5	62.0
Ours - L2R	30.5	25.3	54.5	95.6	40.5	33.7	35.6	17.2	33.2	32.7	54.4	50.2
Ours - Common	28.0	24.8	55.5	90.3	37.1	29.8	33.9	15.0	31.1	28.2	50.8	53.1
Ours - Rare	28.1	24.5	52.9	91.4	31.1	27.9	34.1	15.2	31.3	26.4	48.5	55.1
Ours - VOI	**31.0**	**25.7**	**56.0**	**100.6**	**45.9**	**34.5**	**36.6**	**17.6**	**34.0**	**32.9**	**54.6**	49.3

Table 3.1: Results of MS-COCO, Django, Gigaword, and WMT with fixed orders (L2R, Random, Common, Rare) as baseline. Here, R-1, R-2, and R-L indicate ROUGE-1, ROUGE-2, and ROUGE-L, respectively. For TER, lower is better; for all other metrics, higher is better. "—" = not reported. B, M, C, and A represent BLEU, Meteor, CIDEr, and Accuracy metrics respectively.

3.6 Order Analysis

In this section, we analyze the generation orders learned by *Variational Order Inference* on a macro level by comparing the similarity of our learned orders with predefined orders defined in Section 3.5, and on a micro level, by inspecting when the model generates certain *types* of tokens.

3.6.1 Understanding The Model Globally

We find that prior work [102, 103, 128] tends to study autoregressive orders by evaluating performance on validation sets, and by visualizing the model's generation

[1] For InDIGO-SAO, we report the results on COCO and Django trained using our own implementation. We did not attempt SAO on Gigaword or WMT due to the large dataset sizes, which can take a very long time to train. For WMT, we report the SAO result as in the original paper, and we follow their evaluation scheme.

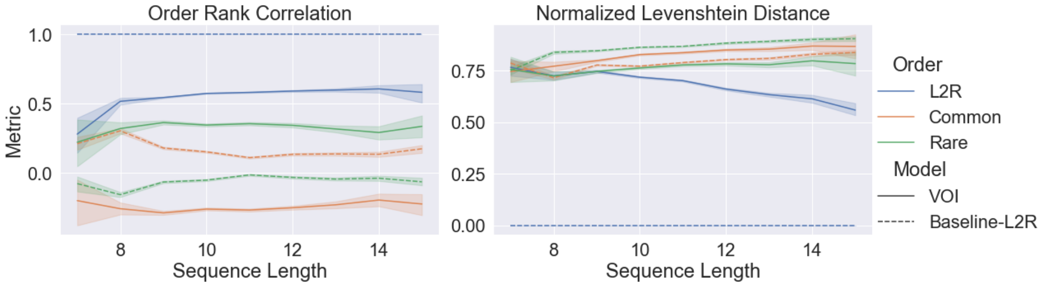


Figure 3.6: **Global statistics for learned orders.** We compare metrics as a function of the sequence length of generated captions on the COCO 2017 validation set. On the left, we compare orders learned with *Variational Order Inference* to a set of predefined orders (solid lines) using *Order Rank Correlation*. As a reference, we provide the *Order Rank Correlation* between L2R and the same set of predefined orders (dashed lines). In the right plot, with identical setup, we measure *Normalized Levenshtein Distance*. We observe that *Variational Order Inference* favors left-to-right decoding above the other predefined orders—this corresponds to the blue lines. However, with a max *Order Rank Correlation* of 0.6, it appears left-to-right is not a perfect explanation. The comparably high *Order Rank Correlation* of 0.3 with rare-tokens-first order suggests a complex strategy.

steps. We provide similar visualizations in Appendix.

$$\mathcal{D}_{NLD}\left(\mathbf{w},\mathbf{z}\right) = \operatorname{lev}\left(\mathbf{w},\mathbf{z}\right)/n \tag{3.5}$$

$$\operatorname{lev}\left(\mathbf{w},\mathbf{z}\right) = 1 + \min\left\{\operatorname{lev}\left(\mathbf{w}_{1:},\mathbf{z}\right),\operatorname{lev}\left(\mathbf{w},\mathbf{z}_{1:}\right),\operatorname{lev}\left(\mathbf{w}_{1:},\mathbf{z}_{1:}\right)\right\} \tag{3.6}$$

The function $\operatorname{lev}\left(\mathbf{w},\mathbf{z}\right)$ is the Levenshtein distance, and $z_{1:}$ removes the first element of z. This metric has the property that a distance of 0 implies that two orders $\mathbf{w}$ and $\mathbf{z}$ are the same, while a distance of 1 implies that the same tokens appear in distant locations in $\mathbf{w}$ and $\mathbf{z}$. Our second metric *Order Rank Correlation*, is the Spearman's rank correlation coefficient between $\mathbf{w}$ and $\mathbf{z}$.

$$\mathcal{D}_{ORC}\left(\mathbf{w},\mathbf{z}\right) = 1 - 6\cdot\textstyle\sum_{i=0}^{n}\left(\mathbf{w}_i - \mathbf{z}_i\right)/\left(n^3 - n\right) \tag{3.7}$$

A correlation of 1 implies that $\mathbf{w}$ and $\mathbf{z}$ are the same; a correlation of -1 implies that $\mathbf{w}$ and $\mathbf{z}$ are reversed; and a correlation of 0 implies that $\mathbf{w}$ and $\mathbf{z}$ are not correlated. In Figure 3.6, we apply these metrics to analyze our models learnt through *Variational Order Inference* .

Discussion. The experiment in Figure 3.6 confirms our model's behavior is not well explained by predefined orders. Interestingly, as the generated sequences increase in length, the *Normalized Levenshtein Distance* decreases, reaching a final value of 0.57, indicating that approximately half of the tokens are already arranged according to a

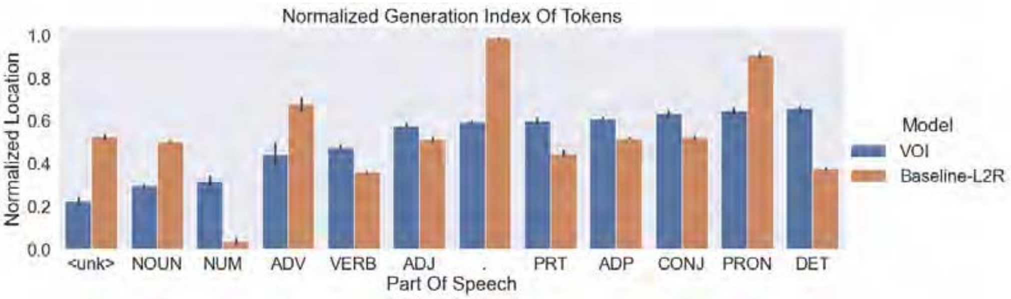


Figure 3.7: **Local statistics for learned orders.** In this figure, we evaluate the normalized generation indices for parts of speech in predicted captions on the COCO 2017 validation set. The normalized generation index is defined as the absolute generation index of a particular token, divided by the final length of predicted sequence. Parts of speech (details in Appendix **??**) are sorted in ascending order of average normalized location. We observe that *modifier* tokens, such as “the”, tend to be decoded last, while *descriptive* tokens, such as nouns and verbs, tend to be decoded first.

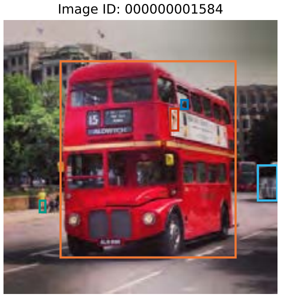
left-to-right generation order. However, the *Order Rank Correlation* barely increases, so we can infer that while individual tokens are close to their left-to-right generation index, their relative ordering is not preserved. Our hypothesis is that certain phrases are generated from left-to-right, but their arrangement follows a *best-first* strategy.

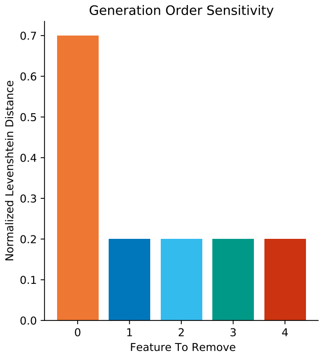
3.6.2 Understanding The Model Locally

To complement the study of our model at a global level, we perform a similar study on the micro token level. Our hope is that a per-token metric can help us understand if and when our *Variational Order Inference* is adaptively choosing between left-to-right and rare-first order. We also hope to evaluate our hypothesis that *Variational Order Inference* is following a *best-first* strategy.

Discussion. The experiment in Figure 3.7 demonstrates that *Variational Order Inference* prefers decoding *descriptive* tokens first—such as nouns, numerals, adverbs, verbs, and adjectives. In addition, the unknown part of speech is typically decoded first, and we find this typically corresponds to special tokens such as proper names. Our model appears to capture the *salient* content first, which is illustrated by nouns ranking second in the generation order statistics. For image captioning, nouns typically correspond to focal objects, which suggests our model has an object-detection phase. Evidence of this phase supports our previous hypothesis that a *best-first* strategy is learned.

3.6.3 Understanding The Model Via Perturbations





In this section, we study the question: to what extent is the generation order learned by *Variational Order Inference* dependent on the content of the conditioning variable **x**? This question is important because simply knowing that our model has learned a *best-first* does not illuminate whether that strategy depends only on the target tokens **y** being generated, or if it also depends on the content of **x**. An adaptive generation order should depend on both.

Discussion. In this experiment, we first obtain a sequence **y** generated by our VOI given the source image **x**. We then freeze **y**, which allows the model to infer a new generation order for **y** when different features of **x** are removed. The right figure shows that for a particular case, removing a single region-feature (feature number 0, which corresponds to the bus) from **x** changes the model-predicted generation order by as much as 0.7 *Normalized Levenshtein Distance*. These results confirm that our model appears to learn an *adaptive* strategy, which depends on both the tokens **y** being generated and the content of the conditioning variable **x**, which is an image in this experiment.

3.7 Ablation Studies

In Section 3.5, we introduced the specific encoder and decoder architectures we use for conditional sequence generation tasks. In this section, we present ablation studies to support the architecture design of our encoder and modeling q_ϕ with Gumbel-Matching distribution.

We consider 4 different positional encoding schemes for the encoder Transformer ϕ: the sinusoid encoding in the original Transformer [112], the sinusoid encoding with positional attention module [128], the relative positional encoding in [129], and the relative positional encoding proposed in Transformer-XL [113]. Besides modeling $q_\phi(\cdot|\mathbf{x}, \mathbf{y})$ as Gumbel-Matching distribution and using Bethe permanent to approximate its denominator, we also consider modeling using Plackett-Luce distribution [130, 131] and sample using techniques recently proposed in [108]. Plackett-Luce distribution has tractable density, so we can compute the exact q_ϕ efficiently without

Enc \ Distrib	Gumbel-Matching	Plackett-Luce
Sinusoid	0.40	0.62
Sinusoid + Pos Attn	0.42	0.58
Relative	0.38	0.53
XL-Relative	**0.25**	0.57

Table 3.2: Normalized Levenshtein Distance between the ordering learnt by the encoder and the ground truth ordering, under different positional encodings (enc) and modeling distributions of q_ϕ (distrib).

using approximation techniques.

To analyze the encoder's ability to learn autoregressive orderings, we first train a decoder with Common-First order on one batch of MS-COCO until it perfectly generates each sentence. We then fix the decoder and initialize an encoder. We train the encoder for 15k gradient steps using the procedure in Algorithm 1 to recover the ground truth Common-First order, and we report the final Normalized Levenshtein Distance against the ground truth in Table 3.2. We observe that modeling q_ϕ with Gumbel-Matching distribution significantly outperforms modeling with Plackett-Luce, despite the former requiring denominator approximation. We also observe that under Gumbel-Matching modeling distribution, the relative position encoding in Transformer-XL significantly outperforms other encoding schemes. Thus we combine these two techniques in our architecture design.

In addition, we analyze how choices of K, the number of latents per training sample, affects model performance. We use the same setting as above and apply Transformer-XL relative position encoding, and we report the results in Table 3.3. We observe that the encoder more accurately fits to the ground truth order as K increases, until a value of around 10. Since a very large K can slow the model down while only bringing marginal improvements, we find a good choice of K to be between 4 and 10.

Table 3.3: Normalized Levenshtein Distance between the encoder ordering and the ground truth with respect to the choice of K.

K	2	3	4	10	20
$\mathcal{D}_{NLD}$	0.31	0.28	0.25	0.21	0.21

3.8 Conclusion

We propose, to our best knowledge, the first unsupervised learner that learns high-quality autoregressive orders through fully-parallelizable end-to-end training without domain-specific tuning. We propose a procedure named *Variational Order Inference* that uses the Variational Lower Bound with the space of autoregressive orderings as latent. Building on techniques in combinatorical optimization, we develop a practical policy gradient algorithm to optimize the encoder of the variational objective, and we propose an encoder architecture that conditions on training examples to output autoregressive orders. Empirical results demonstrate that our model is capable of discovering autoregressive orders that are competitive with or even better than fixed and predefined orders. In addition, the global and local analysis of the orderings learned through *Variational Order Inference* suggest that they resemble a type of *best-first* generation order, characterized by prioritizing the generation of *descriptive* tokens and deprioritizing the generation of *modifier* tokens.

Chapter 4

More Control for Free! Image Synthesis with Semantic Diffusion Guidance

4.1 Introduction

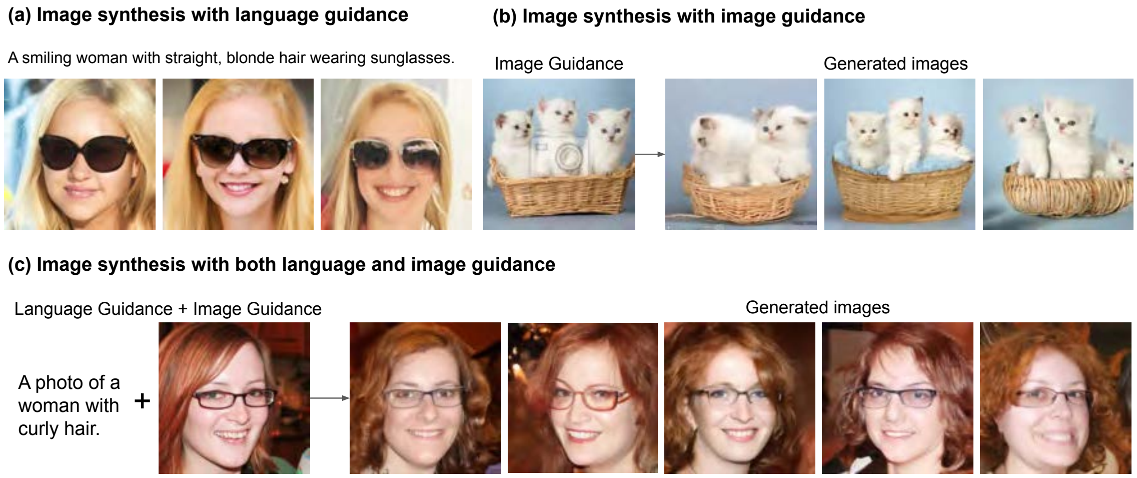

(a) Image synthesis with language guidance
A smiling woman with straight, blonde hair wearing sunglasses.
(b) Image synthesis with image guidance
Image Guidance
Generated images
(c) Image synthesis with both language and image guidance
Language Guidance + Image Guidance
A photo of a woman with curly hair.
+
Generated images

Figure 4.1: We incorporate flexible and lightweight semantic guidance into diffusion models for image synthesis. Our method allows fine-grained semantic control via language guidance, image guidance, or both, and can be applied to datasets without paired image-caption data.

Image synthesis has made great progress in recent years [132–136]. In addition to the goal of generating high-quality photo-realistic images, fine-grained control over the generated images is also an important desideratum when assisting users with art creation and design.

Previous works have explored controllable image synthesis by adding different conditions, including language [135,137,138], attributes [139–141], scene graphs [142], and user sketch or scribbles [143]. Specifically, text-to-image synthesis, as shown in Figure 5.1-(a), aims to generate images based on text instructions, by adding text embeddings as conditional information to the image generation network. However, most previous text-to-image synthesis methods require image-caption pairs for training, and cannot generalize to datasets without text annotations.

Besides text instructions, users may also want to guide the image generation model with a reference image. E.g., a user might want to generate cat images which look similar to a given photo of a cat in terms of its appearance. This information cannot be easily described by language, but can be provided via a reference image, as shown in Figure 5.1-(b). Moreover, sometimes a user may want to provide both language and image guidance. For example, a user might seek to generate "a woman with curly hair" that looks similar to a reference image of a woman with red hair, as illustrated in Figure 5.1-(c).

Current image-conditioned synthesis techniques would either only transfer the "style" of a reference image to a target image [144,145] or are restricted to the domains with a well-defined structure such as human or animal faces [141, 144]. However, they cannot generate diverse images with various pose, structure, and layout based on a single reference image.

We propose *Semantic Diffusion Guidance (SDG)*, a unified framework for text-guided and image-guided synthesis that overcomes these limitations. Our model is based on denoising diffusion probabilistic models (DDPM) [146] which generates an image from a noise map and iteratively remove noise to approach the data distribution of natural images.

We inject the semantic input by using a guidance function to guide the sampling process of an unconditional diffusion model. This enables more controllable generation in diffusion models and gives us a unified formulation for both language and image guidance. Specifically, our language guidance is based on the image-text matching score predicted by CLIP [147] finetuned on noised images. As for the image guidance, depending on what information we seek in the image, we define two options: content and style guidance. The flexibility of the guidance module allows us to inject either language or image guidance alone or both at once into any unconditional diffusion model without the need for re-training. We propose a self-supervised scheme to finetune the CLIP image encoder without text annotations, from which we obtain the guidance model with minimal cost.

Our unified framework is flexible and allows fine-grained semantic control in image synthesis with various applications, as shown in Figure 5.1. We show that our model can handle: (1) Text-guided image synthesis with a complex fine-grained text query on any dataset *without language annotations*; (2) Image-guided image synthesis with content or style control from an input image, which generates diverse images with different pose, structure, and layout; (3) Multi-modal guidance for image synthesis with both language and image input. Our guidance network can be injected into off-the-shelf unconditional diffusion models, *without the need for finetuning or re-training* the diffusion model. We conduct experiments on FFHQ [148] and LSUN [149] datasets to validate the quality, diversity, and controllability of our generated images, and show various applications of our proposed Semantic Diffusion Guidance.

4.2 Related Work

Text-guided Synthesis Pioneered by GAN-INT-CLS [150] and GAWWN [151], conditional generative adversarial networks (GANs) [152] have been the dominant framework for text-based image synthesis. Various methods have been studied, proposing many different text-adaptive architectures and loss functions to enforce better semantic alignment between the input text and generated image, resulting in significant improvements in editing quality and correctness [153–165].

Recent work DALL-E [135] shows promising results with transformers [14] and discrete VAE [166] by leveraging web-scale data. A concurrent work GLIDE [22] adapts classifier-free guidance for large diffusion models and large-scale training for text-guided image synthesis.

Despite great advancements, prior methods require paired image-text annotations which limits the application to certain datasets or requires large amount of data and computational resources for training. Our proposed framework is able to generate images on multiple domains given detailed text prompts, requiring neither image-text paired data from those domains nor large amount of compute to train the text-guided image synthesis model.

Image-guided Synthesis Image-guided synthesis aims at generating diverse images with the constraint that they all should resemble a given reference image in terms of content or style. Many style transfer works fall under this category where the content of the input image must be preserved while the style of the reference image is transferred [167–176], yet they struggle to generate diverse images. Some work investigate image synthesis guided by the content of the refernce images. ILVR [177] proposes a way to iteratively inject image guidance to a diffusion model, yet it exhibits limited structural diversity of the generated images. Instance-Conditioned

GAN [178] utilizes nearest neighbor images of a given reference for adversarial training to generate structurally diverse yet semantically relevant images. Nonetheless, it requires training the GAN model with instance-conditioned techniques. Our approach demonstrates better controllability as different types of image guidance are proposed where users can decide how much semantic, structural, or style information to preserve by using different types and scales of guidance, while not needing to re-train the unconditional diffusion model.

Diffusion Models Diffusion models are a new type of generative models consisting of a forward process (signal to noise) and a reverse process (noise to signal). The denoising diffusion probabilistic model (DDPM) [146, 179] is a latent variable model where a denoising autoencoder gradually transforms Gaussian noise into real signal. Score-based generative model [180, 181] trains a neural network to predict the score function which are used to draw samples via Langevin Dynamics. In [182], it is shown that diffusion probabilistic models and score-based generative models fall under the same framework as both can be viewed as discretizations to stochastic differential equations. Collectively, these models have demonstrated comparable or superior image quality compared to GANs while exhibiting better mode coverage and training stability. Diffusion models have also been explored for conditional generation such as class-conditional generation, image-guided synthesis, super-resolution, and image-to-image translation [136, 177, 182, 183]. Concurrent work [184] explored text-guided image editing with diffusion models. In this work, we further explore whether diffusion models can be semantically guided by text or image, or both to synthesize realistic images.

CLIP-guided Generation CLIP [147] is a powerful vision-language joint embedding model trained on large-scale images and texts. Its representations have been shown to be robust and general enough to perform zero-shot classification and various vision-language tasks on diverse datasets. StyleCLIP [185] and StyleGAN-NADA [148] have demonstrated that CLIP enables text-guided *image manipulation* without domain-specific image-text pairs. However, the application to *image synthesis* has not been explored. Our work investigates text and/or image guided synthesis using CLIP and unconditional DDPM.

4.3 Semantic Diffusion Guidance

We propose *Semantic Diffusion Guidance (SDG)*, a new unified framework that incorporates different forms of guidance into a pretrained unconditional diffusion model. SDG can leverage language guidance, image guidance, and multimodal guidance, enabling controllable image synthesis. The guidance module can be

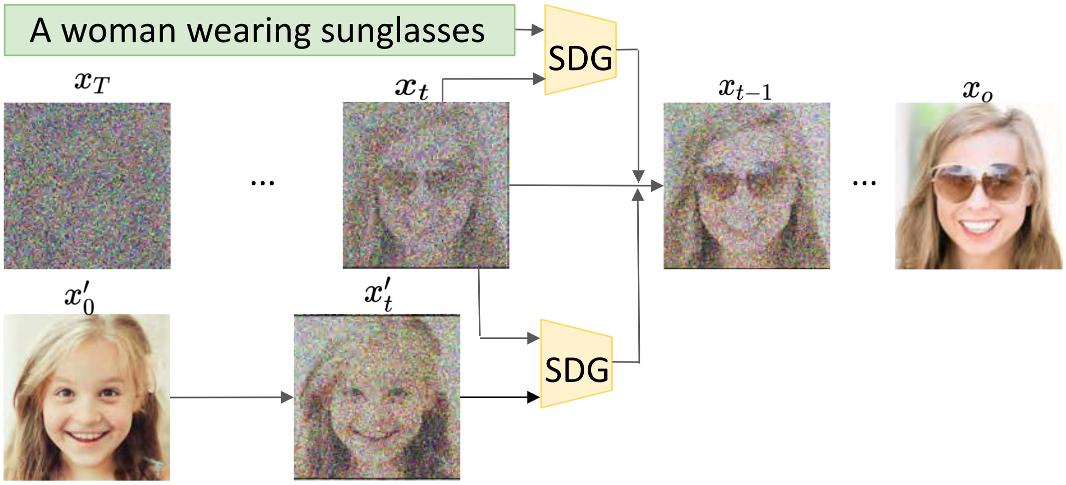


Figure 4.2: An overview our method. Our method is based on the DDPM model which generates an image from a noise map by iteratively removing noise at each timestep. We control the diffusion generation process by Semantic Diffusion Guidance (SDG) with language and/or a reference image. SDG is iteratively injected at each step of generation process. We only illustrate the guidance at one timestep t in the figure.

injected into any off-the-shelf unconditional diffusion model without re-training or finetuning it. We only need to finetune the guidance network, which is a CLIP [147] model in our implementation, on the images with different levels of noise. We propose a self-supervised finetuning scheme, which does not require paired language data to finetune the CLIP image encoder.

In Section 4.3.1, we review the preliminaries on diffusion models, and introduce our approach for injecting guidance into the diffusion model for controllable image synthesis. In Section 4.3.2, we illustrate the language guidance which enables the unconditional diffusion model to perform text-to-image synthesis. In Section 4.3.3, we propose two types of image guidance, which take the content and style information from the reference image as the guidance signal, respectively. In Section 4.3.5, we explain how we finetune the CLIP guidance network without requiring text annotations in the target domain.

4.3.1 Guiding Diffusion Models for Controllable Image Synthesis

Diffusion models define a Markov chain where random noise is gradually added to the data, known as the forward process. Formally, given a data point sampled from a real-data distribution $x_0 \sim q(x)$, the forward process sequentially adds Gaussian noise to the sample over T timesteps:

$$\begin{aligned} q(x_t|x_{t-1}) &= \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}) \\ q(x_{1:T}|x_0) &= \prod_{t=1}^{T} q(x_t|x_{t-1}), \end{aligned} \tag{4.1}$$

where $\{\beta\}_{t=1:T}$ denotes a constant or learned variance schedule that controls the noise step size. A property of the forward process is that we can sample x_t from x_0 in a closed form:

$$q(x_t|x_0) = \sqrt{\bar{\alpha}_t}x_0 + \epsilon\sqrt{1-\bar{\alpha}_t}, \epsilon \sim \mathcal{N}(0,1) \tag{4.2}$$

where $\alpha_t = 1 - \beta_t$ and $\overline{\alpha_t} = \prod_{s=1}^{t} \alpha_s$.

Generative modeling is done by learning the backward process where the forward process is reversed via a parameterized diagonal Guassian transition:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t), \sigma_\theta^2(x_t)\mathbf{I}) \tag{4.3}$$

We choose the notation $p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta, \sigma_\theta^2\mathbf{I})$ for brevity. In order to learn the backward process, neural networks are trained to predict μ_θ and σ_θ^2.

The formulations above explain the unconditional backward process $p_\theta(x_{t-1}|x_t)$; with an extra guidance signal y, the sampling distribution becomes:

$$p_{\theta,\phi}(x_{t-1}|x_t, y) = Z p_\theta(x_{t-1}|x_t) p_\phi(y|x_{t-1}), \tag{4.4}$$

where Z is a normalizing constant. It is proven in [136] that the new distribution after incorporating the guidance can also be approximated by a Gausiann distribution with shifted mean:

$$p_\theta(x_{t-1}|x_t) p_\phi(y|x_{t-1}) = \mathcal{N}(\mu + \Sigma g, \Sigma), \tag{4.5}$$

where $\mu = \mu_\theta$, $\Sigma = \sigma_\theta^2\mathbf{I}$, $g = \nabla_{x_{t-1}} \log p_\phi(y|x_{t-1})$.

Class-guided synthesis was explored in [136] where y is a discrete class label, and $p_\phi(y|x_{t-1})$ is the probability of x_{t-1} belonging to class y. In this work, we generalize y to a continuous embedding for language, image or multimodal guidance. In the next subsections, we introduce how we define the guidance function $F_\phi(x_t, y, t) = \log p_\phi(y|x_t)$ for different guidance.

Figure 4.2 and Algorithm 2 summarize the proposed Semantic Diffusion Guidance. Note that there is an additional scaling factor s for semantic guidance in Algorithm 2 which is a user-controllable hyperparameter that determines the strength of the guidance. We discuss its effect in Section 5.5.

ruled

Algorithm 2 Semantic Diffusion Guidance

```
KwDataInput guidance $y$, scaling factor $s$ **Given:** diffusion model $(\mu_\theta,\ \sigma_\theta)$, Guidance function $F_\phi(x_t, y, t)$
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
**for** $t = T, \cdots 1$ **do**
$\mu, \Sigma \leftarrow \mu_\theta, \sigma_\theta^2 \mathbf{I}$
$x_{t-1} \leftarrow$ sample from $\mathcal{N}(\mu + s\Sigma\nabla_{x_t} F_\phi(x_t, y, t), \Sigma)$
**end for**
**return** $x_0$
```

4.3.2 Language Guidance

Language is one of the most intuitive ways that a user can control the generation model. In order to incorporate language information to the image synthesis process, we use a visual-semantic embedding model for image-text alignment. Specifically, given an image x and a text prompt l, the model embeds them into the joint embedding space using an image encoder E_I and a text encoder E_L, respectively. The similarity between the embeddings $E_I(x)$ and $E_L(l)$ is calculated as the cosine distance, and we utilize this to formulate the language guidance function.

However, note that the models for backward process and guidance in Equation 4.5 are time-dependent, and take noisy images as input. This means that the image encoder E_I needs to incorporate the timestep t as input and be further trained on noisy images at different timesteps as well. We denote such time-dependent image encoder for noisy images as E'_I. Finally, the language guidance function can be defined as:

$$F(x_t, l, t) = E'_I(x_t, t) \cdot E_L(l), \tag{4.6}$$

where $E\prime$ denotes the image encoder trained on noised images with additional timestep input. In Section 4.3.5, we provide details on adapting a pretrained CLIP model [147] to become time-dependent with minimal architecture changes, and present a self-supervised finetuning strategy for noisy images.

4.3.3 Image Guidance

In some cases, an image can convey information that is difficult to express in language. For example, users may want to generate a photo of a cat that looks similar to another cat, or want to generate a photo of a bedroom in the style of Van Gogh's painting "The Starry Night". They may also want to generate realistic images given an emoji or a painting. We thus propose an approach for image-guided diffusion that effectively controls the content or style information according to an image. We present two types of image guidance, namely *image content guidance* and *image style guidance*.

Image Content Guidance aims to control the content of the generated image, with or without structural constraints, based on a reference, and is formulated as the cosine similarity of the image feature embeddings. Let x'_0 denote the noise-free reference image. We perturb x'_0 per Equation 4.2 to get x'_t. Then, the guidance signal at timestep t is,

$$F(x_t, x'_t, t) = E'_I(x_t, t) \cdot E'_I(x'_t, t). \tag{4.7}$$

Similar to language guidance, we use an image encoder finetuned with noised images to define the image guidance function and extract embeddings that mostly capture the high-level semantics. An interesting property of using image encoders for guidance is that one can control how much structural information such as pose and viewpoint is maintained from the reference image. For instance, the embeddings used in Equation 4.7 do not have spatial dimensions, resulting in samples with great variations in pose and layout. However, by utilizing spatial feature maps and forcing alignment between features in corresponding spatial locations, we can guide the generated image to additionally share similar structure with the reference image as follows.

$$F(x_t, x'_t, t) = -\sum_j \frac{1}{C_j H_j W_j} ||E'_I(x_t, t)_j - E'_I(x'_t, t)_j||_2^2 \tag{4.8}$$

where $E'_I()_j \in \mathcal{R}^{C_j \times H_j \times W_j}$ denotes the spatial feature maps of the j-th layer of the image encoder E'_I.

Image Style Guidance allows style transfer from the reference image. It is formulated similarly, except the alignment between the Gram matrices of the intermediate feature maps is enforced:

$$F(x_t, x'_t, t) = -\sum_j ||G'_I(x_t, t)_j - G'_I(x'_t, t)_j||_F^2, \tag{4.9}$$

where $G'_I()_j$ is the Gram matrix [186] of the j-th layer feature map of the image encoder E'_I.

4.3.4 Multimodal Guidance

In some application scenarios, image and language may contain complementary information, and allowing both image and language guidance at the same time provides further flexibility for user control. Our pipeline can easily incorporate both by a weighted sum of the two guidance functions, with their scaling factors as weights.

$$F_{\phi_0}(x_t, y, t) = s_1 F_{\phi_1}(x_t, y, t) + s_2 F_{\phi_2}(x_t, y, t). \tag{4.10}$$

By adjusting the weighting factors of each modality, users can control the balance between the language guidance and image guidance.

4.3.5 Self-supervised Finetuning of CLIP without Text Annotations

CLIP [147] is a powerful vision and language model pretrained on large-scale image-text data. We leverage its semantic knowledge to achieve controllable synthesis for diffusion models. To act as a guidance function, CLIP is expected to handle noised images x_t at any timestep t. We make a minor architectural change to CLIP image encoder E_I to accept an additional input t by converting batch normalization layers to adaptive batch normalization layers, where the prediction of scale and bias terms are conditioned on t. We denote this modified CLIP image encoder as $\widetilde{E_I}$. The parameters of $\widetilde{E_I}$ are initialized by the parameters of a pretrained CLIP model E_I, except for the parameters for predicting the scale and bias of the adaptive batch normalization layers.

To finetune $\widetilde{E_I}$, we propose a self-supervised approach in which the task is to force an alignment between features extracted from clean and noised images. Formally, given a batch of N pairs of clean and noised images $\{x_0^i, x_{t_i}^i\}_{i=1}^N$ where t_i is the timestep sampled for the i-th image that governs the amount of noise, we encode x_0^i and $x_{t_i}^i$ with E_I and $\widetilde{E_I}$, respectively. We rely on the contrastive objective used in CLIP to maximize the cosine similarity of the N positive pairs while minimizing the similarity of the remaining negative pairs. We fix the parameters of E_I and use the contrastive objective to finetune the parameters of $\widetilde{E_I}$. With our finetuned CLIP model, the diffusion model can be guided by image or language information that users provide. Moreover, the CLIP model is finetuned in a self-supervised manner without requiring any language data for the target dataset.

4.4 Experiments

4.4.1 Dataset and Implementation Details

We conduct experiments on FFHQ [148] and LSUN [149] cat, horse, and bedroom subsets. FFHQ dataset contains 70,000 images of human faces. LSUN contains 3 million bedroom images, 2 million horse images, and 1.7 million cat images. We use unconditional DDPMs from [136, 177], and finetune CLIP [147] RestNet 50×16 models on noised images on each dataset with initial learning rate 10^{-4} and weight decay 10^{-3}, with a batch size of 256. On FFHQ dataset, the learning rate decays by a factor of 0.1 every 3,000 iterations, and the model is trained for 14,000 iterations. On LSUN cat, LSUN horse, and LSUN bedroom datasets, the learning rate decays by

Table 4.1: Quantitative evaluation of our proposed SDG and comparison to prior work on FFHQ dataset with image guidance and text guidance. For FID, the lower, the better. For other scores, the higher, the better.

		Quality	Diversity	Correctness (retrieval evaluation)			
		FID	LPIPS	Top 1	Top 5	Top 10	Top 20
Image guidance	ILVR (N=32) [177]	17.15	0.439	0.205	0.416	0.556	0.727
	SDG	**14.37**	**0.583**	**0.520**	**0.742**	**0.816**	**0.906**
Text guidance	StyleGAN +CLIP	57.45	0.578	**0.749**	**0.934**	**0.974**	**0.996**
	SDG	**28.38**	**0.610**	0.553	0.795	0.878	0.947

Table 4.2: Ablation study of our proposed SDG with image guidance. The numbers in the brackets after “SDG” indicates the scaling factor. For FID, the lower, the better. For other scores, the higher, the better.

		Quality	Diversity	Correctness (retrieval evaluation)			
		FID	LPIPS	Top 1	Top 5	Top 10	Top 20
LSUN Cat	SDG (100)	**16.02**	**0.617**	0.178	0.443	0.592	0.766
	SDG (200)	16.23	0.565	**0.278**	**0.533**	**0.738**	**0.880**
LSUN Horse	SDG (100)	**10.30**	**0.597**	0.165	0.418	0.568	0.704
	SDG (200)	11.22	0.585	**0.298**	**0.609**	**0.738**	**0.863**
LSUN Bedroom	SDG (100)	**5.18**	**0.633**	0.364	0.745	0.866	0.942
	SDG (200)	5.19	0.550	**0.445**	**0.805**	**0.900**	**0.951**

a factor of 0.1 every 30,000 iterations, and the model is trained for 100,000 iterations. When synthesizing images with our SDG, the scaling factor is a hyperparameter that we manually adjust for each guidance, which will be discussed in Sec. 4.4.3.

4.4.2 Quantitative Evaluation

Evaluation Setup Since our SDG is the first method that unifies text guidance and image guidance for image synthesis, there is no previous work on image synthesis with both image and language guidance. So we evaluate the language-guided image synthesis and image-guided image synthesis separately in order to compare with previous work. We evaluate the **language-guided** generation on FFHQ dataset. For that we define 400 text instructions based on combinations of gender and face attributes from CelebA-Attributes [187]. For example, “A photo of a smiling man with glasses”. The entire list of text instructions is included in the supplementary material.

Table 4.3: Ablation study of our proposed SDG with language guidance on FFHQ dataset. The numbers in the brackets after "SDG" is the scaling factor. For FID, the lower, the better. For other metrics, the higher, the better.

		Quality	Diversity	Correctness (retrieval accuracy)			
		FID	LPIPS	Top 1	Top 5	Top 10	Top 20
FFHQ	SDG (120)	**19.60**	**0.650**	0.248	0.526	0.654	0.795
	SDG (160)	22.63	0.644	0.263	0.548	0.679	0.801
	SDG (320)	28.38	0.610	**0.553**	**0.795**	**0.878**	**0.947**

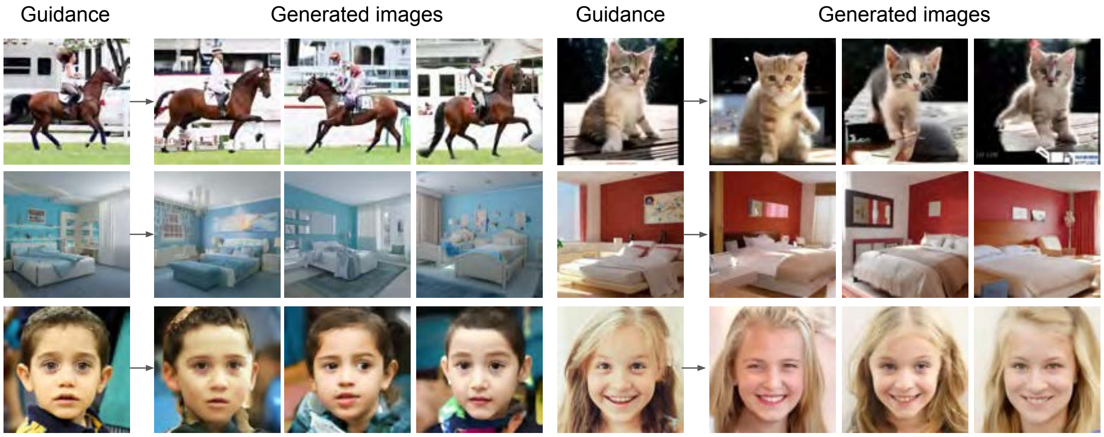


Figure 4.3: Image synthesis results with image content guidance on LSUN and FFHQ datasets. Given a guidance image, the model is able to generate semantically similar images with different pose, layout, and structure.

We generate 25 images for each text query, which results in 10,000 images in total. We compare our language-guided generation with StyleGAN+CLIP[1], which uses CLIP [147] loss to optimize the randomly initialized latent codes of StyleGAN [188] for text-guided image synthesis. Since our model does not require text annotation for training, our text-guided image synthesis experiments are conducted on image-only datasets without paired text annotations. So our method cannot be directly compared with other text-based image synthesis methods which have to be trained on text-image paired datasets. To evaluate **image-guided** image synthesis, we randomly choose 10000 images from the dataset as guidance and synthesize new images based on the guidance images. We compare our image-guided generation results with ILVR [177].

We present quantitative results and comparison with previous work in Table 4.1

[1] https://colab.research.google.com/drive/1br7GP_D6XCgulxPTAFhwGaV-ijFe084X

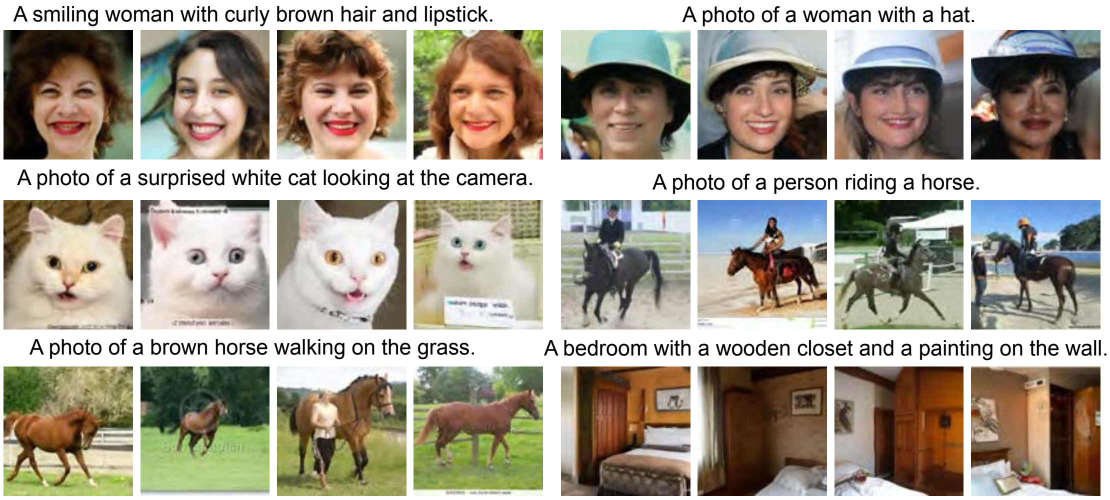

A smiling woman with curly brown hair and lipstick.
A photo of a woman with a hat.
A photo of a surprised white cat looking at the camera.
A photo of a person riding a horse.
A photo of a brown horse walking on the grass.
A bedroom with a wooden closet and a painting on the wall.

Figure 4.4: Image synthesis results with language guidance on LSUN and FFHQ datasets. Our model is able to generate images based on fine-grained language instructions.

with the following evaluation metrics.

FID for image quality evaluation. We report FID score [189] calculated on 10,000 images for each dataset to evaluate the quality of generated images. Lower FID indicates better generation quality. Our SDG outperforms comapred methods for both image-guided synthesis and language-guided synthesis.

LPIPS for diversity evaluation. We calculate the LPIPS score [190] between paired images generated from the same image guidance or the same text guidance, as shown in Table 4.1. Higher LPIPS indicates more diversity. Our model generates more diverse images compared to previous work ILVR [177] and StyleGAN+CLIP. The images generated by ILVR follows the same structure and layout, with variations in details. While our method is able to generate diverse images with different pose, structure, and layout, as shown in Figure 4.7(a). The images generated by StyleGAN+CLIP also suffers from low diversity, as shown in Figure 4.7(b). The high FID score of StyleGAN+CLIP is also because of the low diversity of the generated images.

Retrieval accuracy to evaluate consistency with guidance. We use text-to-image retrieval or image retrieval by an original CLIP ResNet 50×16 model without finetuning to evaluate how well the generated images matches the guidance. For an image generated with text guidance, we randomly select 99 real images from the training set as negative images, and evaluate the text-to-image retrieval performance. Similarly, for an image synthesized with a reference image, we use the reference

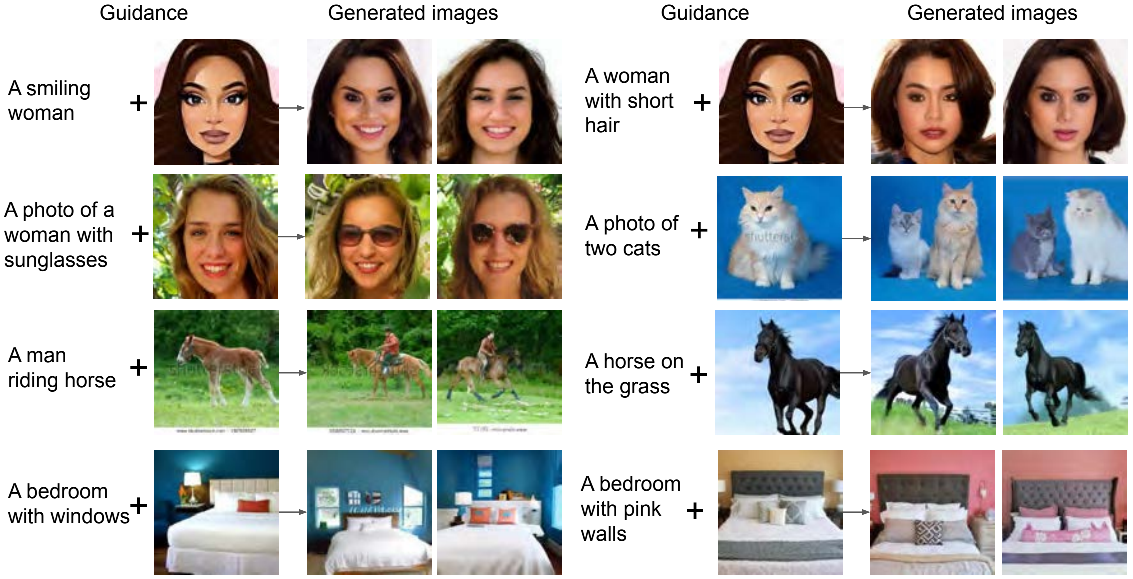


Figure 4.5: Image synthesis results with both image and language guidance. The image and language guidance provides complementary information, and our model generates images that matches both sources of guidance.

image to retrieve the generated image from the 99 randomly selected real images[2]. StyleGAN+CLIP has a very high retrieval performance because the latent codes of the StyleGAN model are directly optimized to minimize the CLIP score calculated by the CLIP model used for retrieval. So the high retrieval performance of StyleGAN+CLIP comes at the cost of low generation diversity, as indicated by the high FID and low LPIPS scores.

4.4.3 Ablation Study

As demonstrated in in Section 4.3.1 and Algorithm 2, the scaling factor s is a user-controllable hyper-parameter that controls the strength of the guidance. We explore the effect of the scaling factor in Table 4.2 and Table 4.3. Figure 4.6 shows the qualitative results for different scaling factors. We observe the trade-off between semantic correctness and diversity of generated images. As the scaling factor gets larger, the guidance signal has more control on the generation results, as indicated by the increased semantic consistency with the guidance. While larger scaling factor

[2] The selected negative images are disjoint with the guidance images we used for synthesizing images.

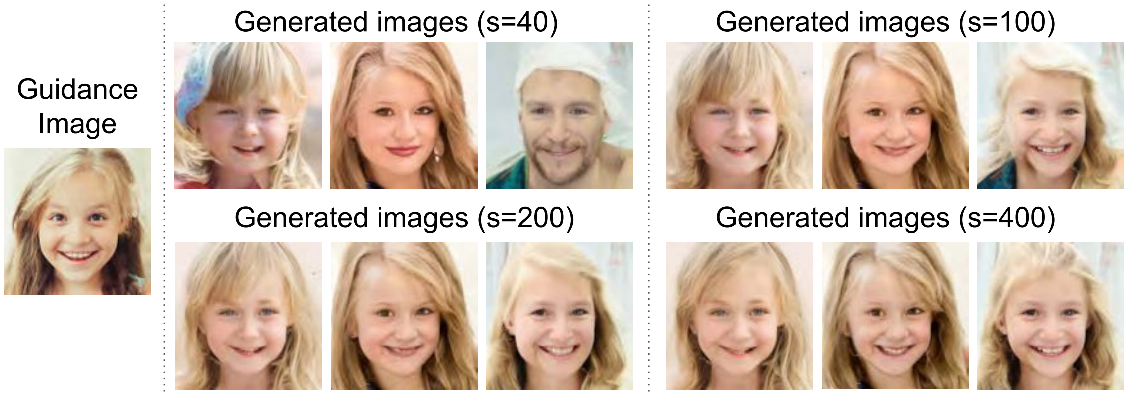


Figure 4.6: Image synthesis results with different scaling factors (s denotes the value of the scaling factor). Larger scaling factors result in lower diversity and more consistency with the guidance.

also leads to lower diversity of generated images. Users can adjust the scaling factor to control how diverse they expect the generated images to be.

4.4.4 Qualitative Results

Text-guided and image-guided synthesis results Our model combines the language and image guidance in a unified framework, and is easy to adapt to various applications. In Figure 4.3 we show the synthesis results with image content guidance (Equation 4.7). With the image guided diffusion, the model is able to synthesize new images with diverse structures that match the semantics of the guidance image. Figure 4.4 shows the language-guided diffusion results, where our model is able to handle complex and fine-grained descriptions, such as “A smiling woman with curly brown hair and lipstick.”, or “A bedroom with a wooden closet and a painting on the wall.” We can also incorporate language and image guidance jointly, as shown in Figure 4.5. The image and language guidance provide complementary information, and our semantic diffusion guidance is able to generate images that align with both. For example, we can generate a bedroom similar to the guidance bedroom image but with windows, or generate a woman according to a guidance image but with a new attribute defined the language guidance (e.g., “smiling” or “short hair” or “sunglasses”).

Comparison to prior work Since there is no prior work that incorporates text and image guidance in the same unified framework, we compare our approach to previous text-guided and image-guided synthesis work. In image-guided synthesis, the most

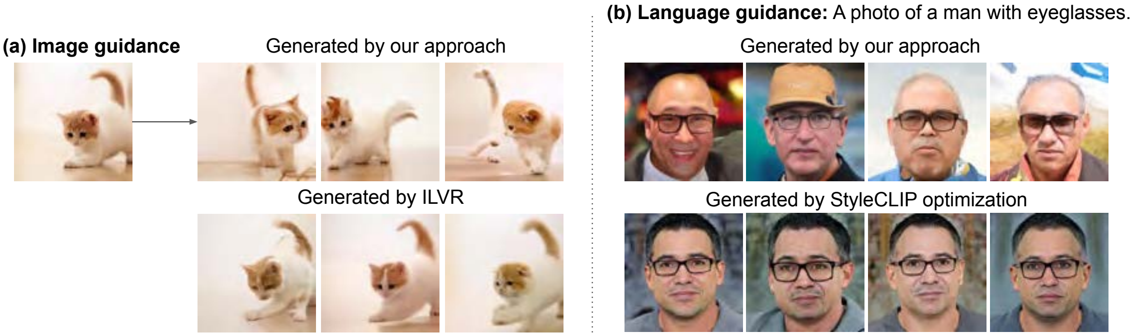

(a) Image guidance
Generated by our approach
Generated by ILVR
(b) Language guidance: A photo of a man with eyeglasses.
Generated by our approach
Generated by StyleCLIP optimization

Figure 4.7: Comparison to previous work. (a) Image-guided image synthesis is compared with ILVR, (b) text-guided image synthesis is compared with StyleGAN+CLIP

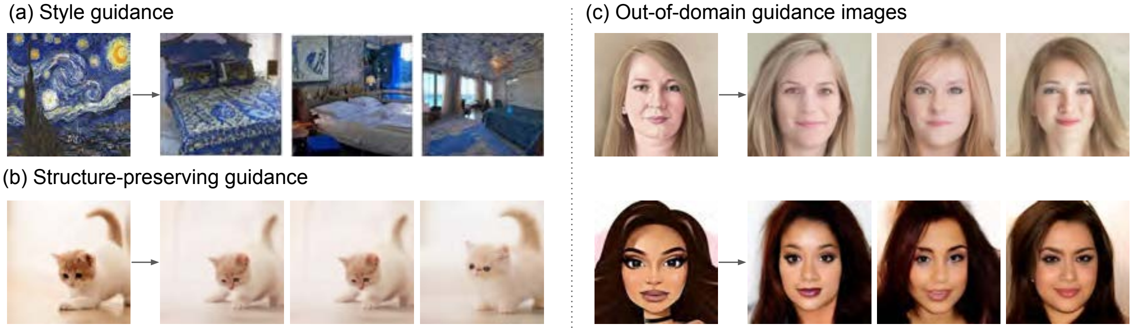

(a) Style guidance
(b) Structure-preserving guidance
(c) Out-of-domain guidance images

Figure 4.8: Different applications of our SDG model. (a) shows style-guided image synthesis. (b) shows structure-preserving image synthesis when the user does not want to generate diverse structures. (c) shows synthesizing realistic images with out-of-domain image guidance.

related to our work is ILVR [177].As shown in Fig. 4.7(a), our model can generate images in different poses and structures, while ILVR can only generate images of the same pose and structure. We compare our language-guided image synthesis with StyleGAN+CLIP in Fig. 4.7(b). Although StyleGAN+CLIP is able to generate high-quality images, diversity is lacking in their results, while our model is able to generate high-quality and diverse results based on the language instructions.

Other applications In Fig. 4.8(a,b), we demonstrate the results of style (Equation 4.9) and structure-preserving (Equation 4.8) image guidance. With the style guidance, the model trained on LSUN bedroom is able to synthesize bedrooms in the unseen style of the reference image. With the structure-preserving content guidance, the synthesized images preserve the structure, pose, and layout information from the reference image. Fig. 4.8(c) shows that the model is able to take an out-of-domain image as guidance, and synthesize photo-realistic images which are semantically similar to the guidance image.

4.5 Conclusion

We propose Semantic Diffusion Guidance (SDG), a unified framework for language-guided and image-guided image synthesis. The semantic diffusion guidance is injected into pretrained diffusion models without extra cost of re-training, and enables fine-grained control over image synthesis with image or language guidance, or both. However, image generation have as much potential for misuse in application as they have for beneficial application. We should be aware of the potential negative social impact if image synthesis is used for generating fake images to mislead people. Please refer to the supplementary materials for more discussion on limitations, failure cases, and potential negative social impacts.

Chapter 5

Shape-Guided Diffusion with Inside-Outside Attention

5.1 Introduction

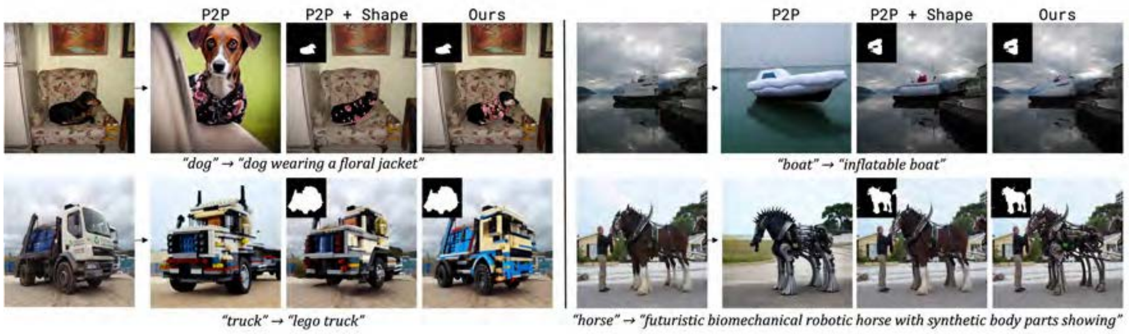

P2P
P2P + Shape
Ours
P2P
P2P + Shape
Ours
"dog" → "dog wearing a floral jacket"
"boat" → "inflatable boat"
"truck" → "lego truck"
"horse" → "futuristic biomechanical robotic horse with synthetic body parts showing"

Figure 5.1: We demonstrate the importance of using an explicit shape when performing a local edit on a real image. Prior work (P2P [5]) has difficulty preserving the source object's shape, even when adapted for local editing (P2P + Shape). We propose *Shape-Guided Diffusion*, a training-free method that uses a novel *Inside-Outside Attention* mechanism to delineate which spatial regions are object vs. background and ensure that edits are localized to the correct region. Our method can be provided an object mask as input or infer a mask from text, as is shown in the above example.

Recent large-scale diffusion models [23, 191, 192], have significantly improved the realism of text-conditional image synthesis and its faithfulness to the input prompt. However, there is a limit to what can be expressed via language. For example, users must perform extensive prompt tuning to achieve a desired silhouette or select one object instance out of many, when their intent could be more easily specified with an object mask. Whether this mask is user specified or implicitly inferred, prior work in image editing is often insensitive to the source object's shape and violates affordances (e.g., producing a dog with missing limbs or a truck with a missing cargo container) or interactions (e.g., producing a boat with an inconsistent reflection) that were present in the original image (see Figure 5.1). Enabling text-to-image diffusion models to respect shape guidance is especially beneficial for applications like anonymization, targeted ads customization, or synthetic data generation. Thus, we consider the task of *shape-guided editing*, where a real image, text prompt, and object mask are fed to a pre-trained text-to-image diffusion model to synthesize a new object faithful to the the text prompt and the mask's shape.

Our method is motivated by the observation that diffusion models often contain spurious attentions that weakly associate object and background pixels. To overcome this issue, we delineate the object (inside) and background (outside) with a novel Inside-Outside Attention mechanism that removes spurious attentions during both the inversion and generation process. This mechanism modifies the cross- and self-attention maps such that a token or pixel referring to the object is constrained to attend to pixels inside the shape, and vice versa.

To summarize, our contributions include the following:
(1) We identify a limitation in prior image editing methods where the shape of the original object is not preserved and provide empirical insights on why this issue exists.
(2) Unlike existing mask-based editing adaptations (e.g., copying the background or finetuning the model to use mask input), we introduce a training-free mechanism that applies a shape constraint on the attention maps at inference time. To the best of our knowledge, we are the first work to explore *constraining* attention maps during inversion, which allows us to discover inverted noise that better preserves shape information from a real image.
(3) Our method achieves SOTA results in shape faithfulness on our MS-COCO ShapePrompts benchmark, and is rated by annotators as the best editing method 2.7x more frequently than the most competitive baseline. We demonstrate diverse editing capabilities such as object edits, background edits, and simultaneous inside-outside edits.

5.2 Related Work

Diffusion Models Diffusion models [179] have had remarkable success in image synthesis. They define a Markov chain of diffusion steps that slowly adds random

Approach	(a) Guidance	(b) Attn Map	(c) Inversion
SDEdit [8]	edit prompt	N/A	N
P2P [5]	src prompt, edit prompt	Copy	Y
InstructPix2Pix* [198]	edit instruction	Copy	N
NTI* [199]	src prompt, edit prompt	Copy	Y
PNP* [200]	src prompt, edit prompt	Copy	Y
Ours	src prompt, edit prompt, shape	Constrain	Y

Table 5.1: A conceptual comparison of our work vs. structure preserving methods. We compare against SDEdit and P2P in a large-scale evaluation, whereas for concurrent works (denoted by *) we include examples in the Supplemental.

noise to data then learn a model that can reverse the diffusion process to construct desired data samples from the noise. Variants of diffusion models include Denoising Diffusion Probabilistic Models (DDPM) [146], Denoising Diffusion Implicit Models (DDIM) [193], and score-based models [182]. Classifier guidance [194] and classifier-free guidance [195] have been investigated for conditional image synthesis. Recently, diffusion models [135,191,192,196] have shown impressive performance on text-guided image synthesis. Our work focuses on adapting these diffusion models towards text-guided local editing according to a text prompt and object mask.

Global and Local Image Editing Researchers have proposed a variety of methods to extend generative models towards image editing. For text-guided global editing, StyleCLIP [185] adapts StyleGAN [188] and DiffusionCLIP [197] adapts diffusion models to edit entire images according to a text prompt using CLIP [147]. Blended Diffusion [6] proposes a method for local editing constrained to a mask by copying an appropriately noised version of the source image's background at each diffusion timestep. While this "copy background" technique can be generally combined with other methods to enable local editing in diffusion models, we demonstrate that this method alone is insufficient for preserving object shape, and we further improve shape faithfulness with our proposed method.

Structure Preserving Image Editing Aside from global and local image editing, there also exists work in structure preserving image editing. These works aim to maintain structure, including the position or pose of the object to be edited. To achieve structure preservation, some works copy random seeds [201], finetune model weights [**?**, 202], copy features and self-attention maps [200], or condition on a partially noised version of the source image [8]. Prompt-to-Prompt (P2P) [5] copies cross-attention maps from the source to target image, and follow up works concurrent to ours improve its performance on real image editing [198, 199].

We present a conceptual comparison of our work vs. a few structure preserving works with open source code in Table 5.1 (see additional examples in the Supplemental). (a) While these methods are often able to produce a background that looks similar to that of the source image, they struggle to perform a local edit where the background is not disturbed because they lack shape as an explicit form of guidance. (b) Unlike prior work that leverages attention maps for image editing, we do not *copy* these attention maps but rather *constrain* them to be sensitive to shape. While directly copied attention maps are noisy and entangle changes in object and background pixels, our constrained attention maps spatially localize these changes, which allows us to perform shape-guided edits (see Figure 5.4). (c) Although P2P demonstrates success in structure consistency when generating multiple synthetic images, it has difficulty preserving the structure of a real image. It shows initial results for real image editing using DDIM inversion [193, 194], a deterministic technique that inverts a real image into noise that would reconstruct the image when fed to the diffusion model for generation. While the inverted noise retains some structure information, combining inversion with classifier-free guidance often causes a drift issue where it is difficult to simultaneously preserve the structure and respect the text prompt. As seen in Figure 5.4, we demonstrate that our shape constraint on the attention maps is able to mitigate this drift issue, which is also explored in the concurrent work Null Text Inversion [199] that instead proposes test-time optimization of null embeddings.

Image Inpainting Image inpainting is the task of generating missing regions of an image for object removal, image restoration, etc. Researchers have proposed dilated convolution [203], partial convolution [204], gated convolution [205], contextual attention [206], and co-modulation [207] for GAN-based image inpainting. Lugmayr *et al*. [208] recently proposed a diffusion-based model for free-form image inpainting. There exist variants of GLIDE [196] and Stable Diffusion [7, 191] finetuned for text-conditional inpainting. However, these methods were trained with free-form masks without semantic meaning, where infilling the mask with background is reasonable and even encouraged. There exist a few training-based methods that use object masks, none of which are publicly available. Make-a-Scene [24] trained an auto-regressive transformer conditioned on full segmentation maps of a scene. Shape-guided Object Inpainting [209] trained a GAN and Imagen Editor [210] trained Imagen [192] with object masks for inpainting. In contrast, we apply our model on top of an open-source text-to-image diffusion model at inference time. Because our method is training-free, it is more flexible and can be applied towards tasks beyond object editing, such as background editing or simultaneous inside-outside editing, as discussed in Sec. 5.5.2.

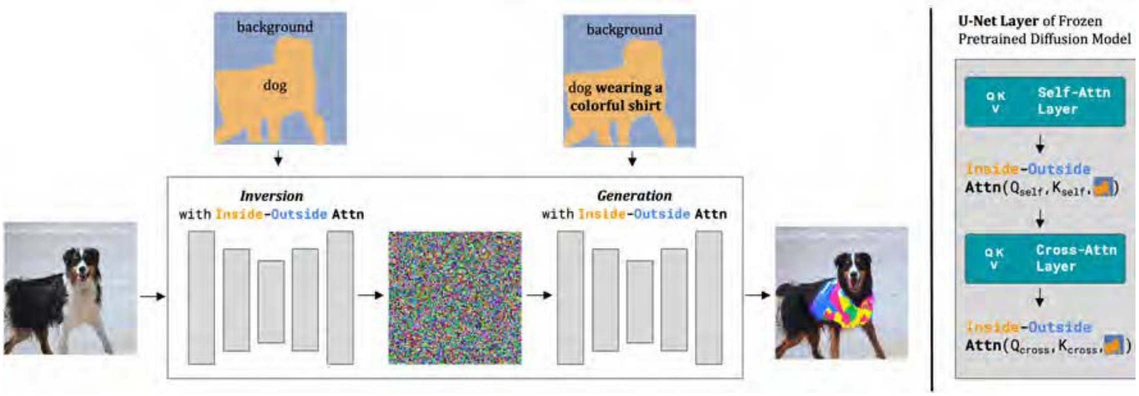


Figure 5.2: Shape-Guided Diffusion. Our method takes a real image, source prompt ("dog"), edit prompt ("dog wearing a colorful shirt"), as well as an optional object mask (inferred from the source prompt if not provided), and outputs an edited image. Left: we modify a frozen pretrained text-to-image diffusion model during both the inversion and generation processes. Right: we show a detailed view of one layer in the U-Net, where Inside-Outside Attention constrains the self- and cross-attention maps according to the mask.

5.3 Shape-Guided Diffusion

We present Shape-Guided Diffusion, a *training-free* method that enables a pretrained text-to-image diffusion model to respect shape guidance. Our goal is to locally edit image x_{src} given text prompts $\mathcal{P}_{src}$ and $\mathcal{P}_{edit}$ and optional object mask m (inferred from $\mathcal{P}_{src}$ if not provided), so that edited image x_{edit} is faithful to both $\mathcal{P}_{edit}$ and m. We introduce Inside-Outside Attention to explicitly constrain the cross- and self-attention maps during both the inversion (image to noise) and generation (noise to image) processes. An overview of our method can be found in Figure 5.2.

We build upon Stable Diffusion (SD), a Latent Diffusion Model (LDM) [191] that operates in low-resolution latent space. LDM latent space is a perceptually equivalent downsampled version of image space, meaning we are able to apply Inside-Outside Attention in latent space via downsampled object masks. For the rest of this paper, when we denote "pixel", "image", or "noise", we are referring to these concepts in LDM latent space.

5.3.1 Inside-Outside Attention

LDMs contain both cross-attention layers used to produce a spatial attention map for each textual token and self-attention layers used to produce a spatial attention

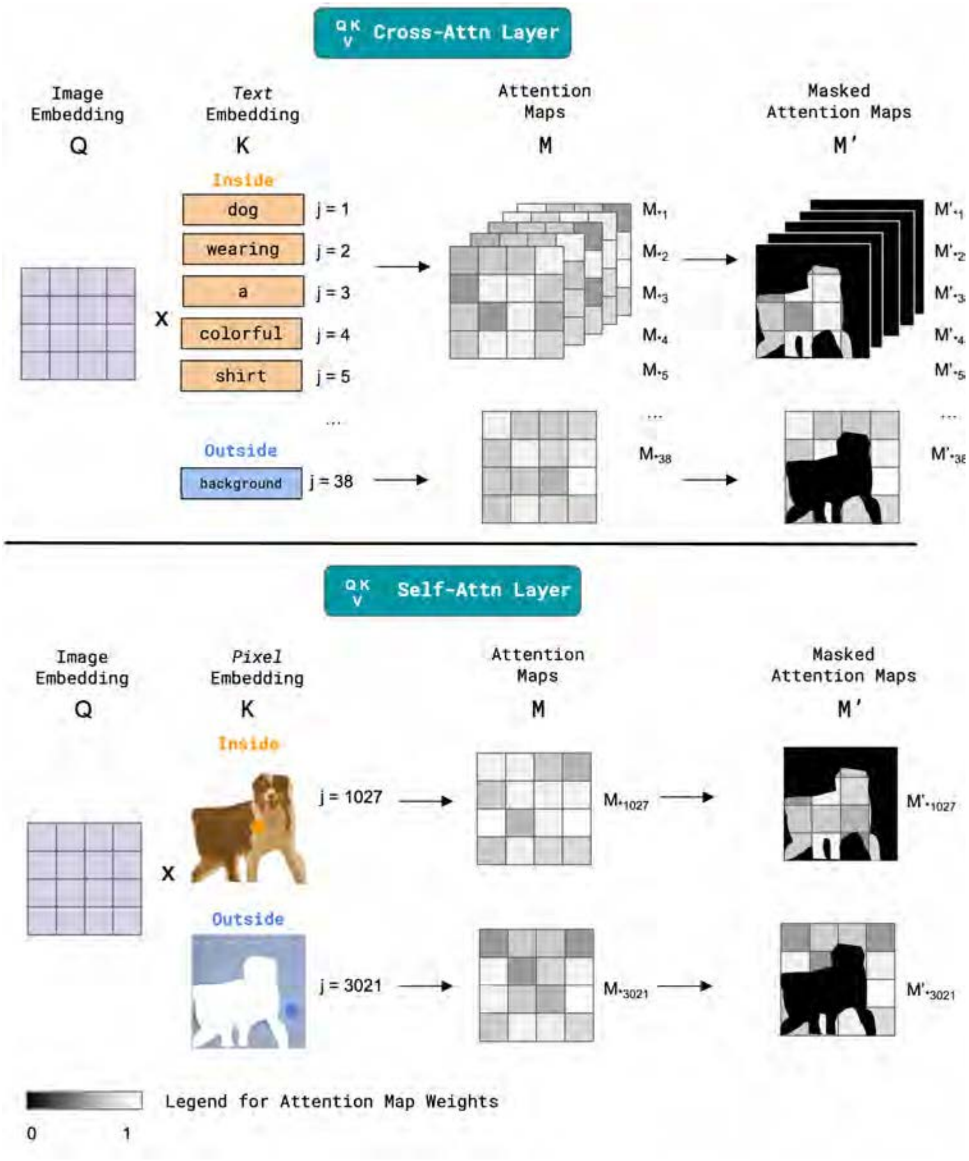


Figure 5.3: Inside-Outside Attention. We modify the attention maps from both the cross-attention and self-attention layers. Here j refers to token/pixel indices and M_{*j} denotes the attention map corresponding to the j-th index. Top: in the cross-attention layer depending on whether the text embedding refers to the inside or outside the object, we constrain the attention map M according to the object mask or the inverted object mask to produce M'. Bottom: in the self-attention layer we perform a similar operation on the inside and outside pixel embeddings.

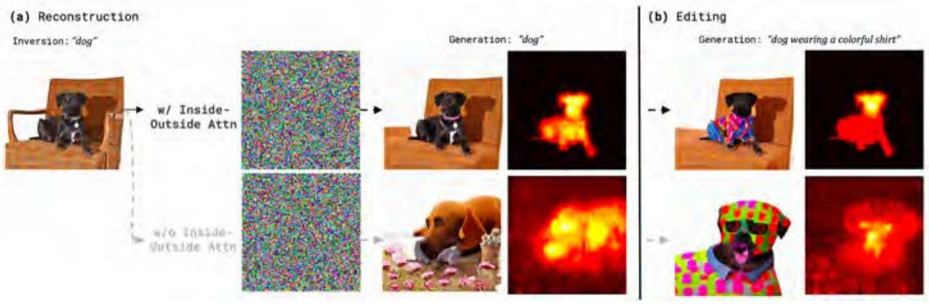


Figure 5.4: Spurious attentions and classifier-free guidance limits shape preservation. Inside-Outside Attention (top) preserves the shape relationship between the object and background by associating tokens to specific spatial regions. We demonstrate this property when reconstructing (left) and editing (right) a real image with classifier-free guidance. We also depict the cross attention map for the token "dog" averaged all attention heads and timesteps.

map for each pixel. We postulate that prior methods often fail because of spurious attentions – attentions that seek to edit the object compete with those that seek to preserve the background because they are not well localized (see Figure 5.4). Hence, we manipulate the cross-attention map such that the inside tokens are responsible for editing a distinct, non-overlapping spatial region compared with the outside tokens (e.g., "dog","shirt", etc. may only edit the dog and "background" may only edit the remaining scene). Since self-attention layers heavily influence how pixels are grouped to form coherent objects, we apply a similar manipulation to the self-attention map to further ensure that the desired object is contained within the boundaries of the input mask.

An overview of Inside-Outside Attention is given in Figure 5.3 and our algorithm is defined as follows (also see Alg. 3). For one forward pass at each timestep during inversion or generation, we go through all layers of the diffusion model DM and manipulate the cross- and self-attention maps M. We denote the dimensions of M as $\mathbb{R}^{HW \times d_\tau}$ and $\mathbb{R}^{HW \times HW}$ for each cross- and self-attention map, respectively, where H is the image height, W is the image width, HW is the number of pixels in the flattened image, and d_τ is the number of tokens. We also downsample m according to the resolution of the cross- or self-attention layer. For the cross-attention map, we determine column indices J_{in} and J_{out} based on whether the token refers to the object or the background. For the self-attention map, we determine column indices J_{in} and J_{out} based on whether the pixel belongs inside or outside the object as defined by mask m. Finally, we compute the new constrained attention maps $M'_{*j_{in}} = \{M_{*j_{in}} \odot m \mid \forall j_{in} \in J_{in}\}$ and $M'_{*j_{out}} = \{M_{*j_{out}} \odot (1-m) \mid \forall j_{out} \in J_{out}\}$.

Algorithm 3 Inside-Outside Attention

Input: A diffusion model DM, a binary object mask m, a prompt $\mathcal{P}$.
Output: An edited diffusion model where the attention maps M are masked according to m and $\mathcal{P}$ for one forward pass.

```
for all $l \in$ layers($DM$) do
    if type($l$) is CrossAttention
        $J_{in} \leftarrow \{j \mid j$th token refers to object$\}$
        $J_{out} \leftarrow \{j \mid j$th token refers to background$\}$
    elif type($l$) is SelfAttention
        $J_{in} \leftarrow \{j \mid j$th pixel belongs inside object$\}$
        $J_{out} \leftarrow \{j \mid j$th pixel belongs outside object$\}$
    $M'_{*j_{in}} = M_{*j_{in}} \odot m \quad \forall j_{in} \in J_{in}$
    $M'_{*j_{out}} = M_{*j_{out}} \odot (1 - m) \quad \forall j_{out} \in J_{out}$
end for
```

5.3.2 Inside-Outside Inversion

To edit real images, we use DDIM inversion [193, 196] to convert the source image to inverted noise. However, we observe that using inversion with a text-to-image diffusion model often results in a shape-text faithfulness tradeoff. Nonzero levels of classifier-free guidance can completely destroy information about the source object (see bottom row of Figure 5.4). We propose applying Inside-Outside Attention to mitigate this trade-off. Similar to how prior work can associate tokens to *entire images* [202, 211], with Inside-Outside Attention we can associate tokens to *specific spatial regions*. As such, if the token remains the same its associated region should remain the same (e.g., “dog” and “background” in the reconstruction setting) and if it changes its associated region should change (e.g., “dog” in the editing setting) without affecting other regions. While without our method (bottom) the inverted noise is able to retain some information about the real image – the checkerboard pattern on the chair is converted to flowers or polka dots in the bottom row – with our method (top) the edited image is able to retain the full chair. We also depict the cross-attention map for “dog”, where without our method the attention map weakly includes the background and with our method the map is localized to the dog.

5.3.3 Method Summary

In summary, we make the observation that object shape can be better preserved if spurious attentions are removed, and we propose the novel inference-time mechanism

Inside-Outside Attention. Our method Shape-Guided Diffusion uses Inside-Outside Attention to constrain the attention maps during both inversion and generation, which we depict in Figure 5.2. The Shape-Guided Diffusion algorithm can be defined as follows (also see Alg. 4).

Algorithm 4 Shape-Guided Diffusion

Input: A diffusion model DM with autoencoder $\mathcal{E}, \mathcal{D}$, real image x_{src}, a source prompt $\mathcal{P}_{src}$, an edit prompt $\mathcal{P}_{edit}$, and either a binary object mask m or a shape inference function InferShape$(\cdot)$.

Hyperparameters: Classifier-free guidance scale w_g.

Output: An edited image x_{edit} that differs from x_{src} only within the mask region m.

```
 1: if $m$ is not provided then
 2:     $m \leftarrow \text{InferShape}(x_{src}, \mathcal{P}_{src})$
 3: end if
 4: $[\bar{z}_0, ..., \bar{z}_T] \sim \text{InsideOutsideInv}(z|\mathcal{E}(x_{src}), \mathcal{P}_{src}, m, DM)$
 5: $z_T \leftarrow \bar{z}_T$
 6: for all $t$ from $T$ to 1 do
 7:     $\text{InsideOutsideAttention}(DM, \mathcal{P}_{edit}, m)$
 8:     $z_{cond} \leftarrow DM(z_t, \mathcal{P}_{edit})$
 9:     $z_{uncond} \leftarrow DM(z_t, \varnothing)$
10:     $z_{t-1} \leftarrow z_{cond} + w_g * (z_{cond} - z_{uncond})$
11:     $z_{t-1} \leftarrow z_{t-1} \odot m + \bar{z}_{t-1} \odot (1 - m)$
12: end for
13: $x_{edit} \leftarrow \mathcal{D}(z_0)$
```

If the mask is not provided, we use the shape inference function InferShape$(\cdot)$ to identify $\mathcal{P}_{src}$ in the image. For our experiments we use an off-the-shelf segmentation model [212], but any method for textual grounding could also be used with our method. We run Inside-Outside Inversion on the conditional diffusion model driven by the prompt $\mathcal{P}_{src}$ (e.g., "dog") to get inverted noise $\bar{z}_T$. We then set our initial noise z_T to $\bar{z}_T$. For each sampling step, we apply Inside-Outside Attention for both the conditional and unconditional diffusion models using mask m and $\mathcal{P}_{edit}$ (e.g., "dog wearing a colorful shirt").

We mix the predictions of both models using the original formulation of Ho *et al.* [146], which applies classifier-free guidance to the conditional prediction (Line 10, Alg. 4). In early experiments we found this design choice leads to higher text alignment without a loss in other metrics. Finally, we copy the real image's background found during the inversion process $\bar{z}_{t-1} \cdot m$ to form the edited image prediction z_{t-1}. This ensures the edited image x_{edit} and the original image x_{src} only differ within the mask region m.

5.4 MS-COCO ShapePrompts

Benchmark We evaluate our approach on MS-COCO images [213]. We filter for object masks with an area between [2%, 50%] of the image, following prior work in image inpainting [214]. Our test set derived from MS-COCO val 2017 contains $1,149$ object masks spanning 10 categories covering animal, vehicle, food, and sports classes. We create a validation set with $1,000$ object masks in the same fashion derived from MS-COCO train 2017. For each category we design a few prompts that add clothing or accessories (e.g., “floral shirt” or “sunglasses”), manipulate color (e.g., “iridescent”, “with spray paint graffiti”), switch material (“lego”, “paper”), or specify rare subcategories (“spotted leopard cat”, “tortilla wrapped sandwich”). More information about the prompts can be found in the Supplemental.
Metrics Since we aim to synthesize an image faithful to the input shape, we use mean Intersection over Union (mIoU) as a metric. Specifically, we compute the proportion of pixels within the masked region correctly synthesized as the desired object class, as determined by a segmentation model [212] trained on COCO-Stuff [215]. Since animal object masks are particularly fine-grained, and mIoU does not capture a full picture of degenerate cases (e.g., if the edit replaces a cat’s full body with a cat’s head), we also compute a keypoint-weighted mIoU (KW-mIoU) for the animal classes. Specifically, we weight each sample’s mIoU by the percentage of correct keypoints when comparing the source vs. edited image, as determined by an animal keypoint detection model [216]. We also report FID scores as a metric for image realism, which measures the similarity of the distributions of real and synthetic images using the features of an Inception network [217, 218]. Finally, we report CLIP [147] scores as a metric for image-text alignment, which measures the similarity of the text prompt and synthetic image using the features of a large pretrained image-text model. More information on metrics can be found in the Supplemental.

5.5 Experiments

In Sec. 5.5.1 we evaluate our method on the shape-guided editing task where it must replace an object given a (real image, text prompt, object mask) triplet from MS-COCO ShapePrompts. We also evaluate on the same task with masks inferred from the text and ablate the use of our Inside-Outside Attention mechanism. In Sec. 5.5.2 we present additional results beyond object editing.

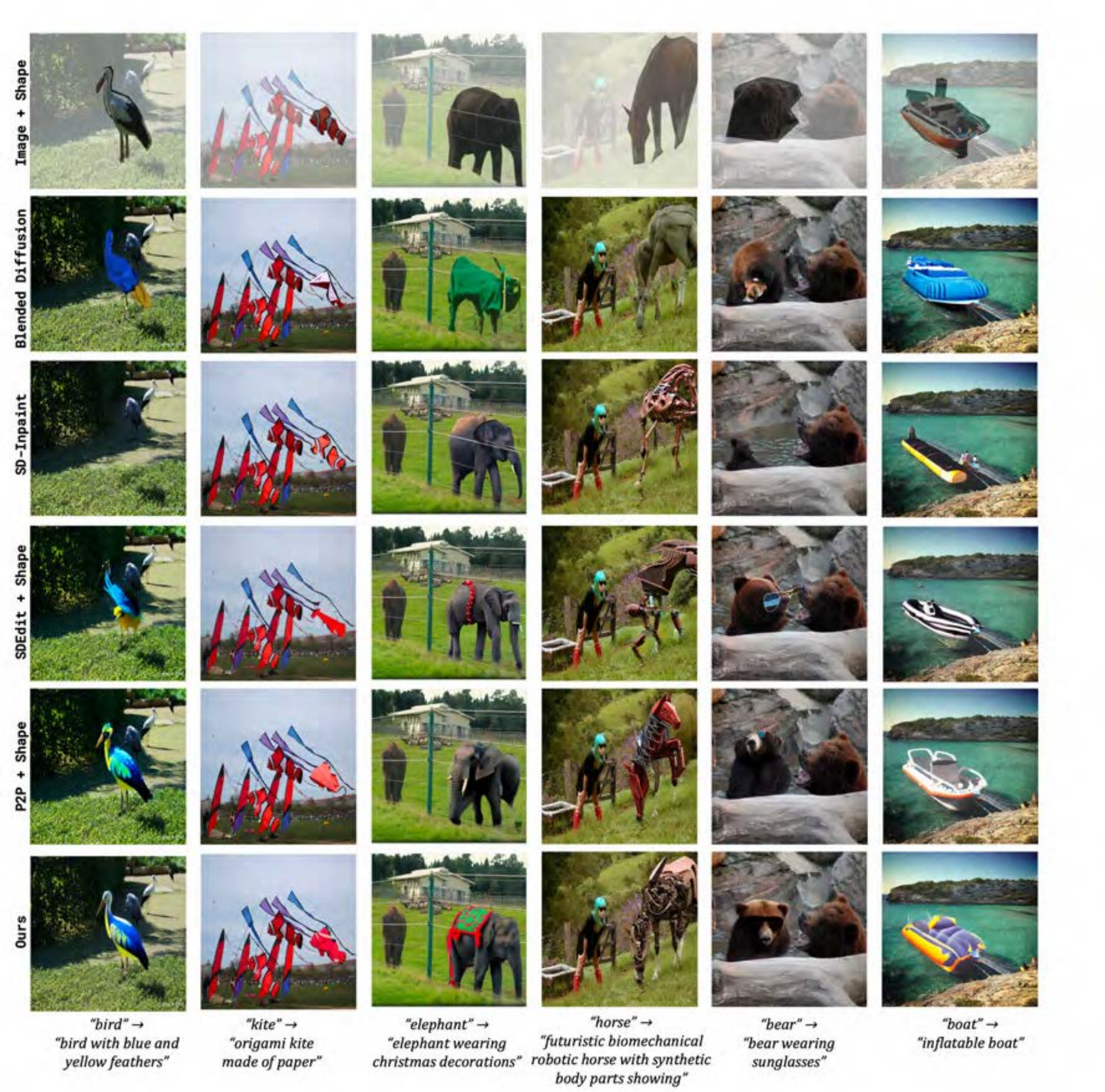


Image + Shape
Blended Diffusion
SD-Inpaint
SDEdit + Shape
P2P + Shape
Ours
"bird" → "bird with blue and yellow feathers"
"kite" → "origami kite made of paper"
"elephant" → "elephant wearing christmas decorations"
"horse" → "futuristic biomechanical robotic horse with synthetic body parts showing"
"bear" → "bear wearing sunglasses"
"boat" → "inflatable boat"

Figure 5.5: Comparison to prior work. We compare our results with Blended Diffusion [6], SD-Inpaint [7], SDEdit [8], and P2P [5] with the MS-COCO image and instance mask for reference. Our method is able to generate realistic edits that are faithful to both the input shape and text prompt. + Shape denotes a variant of the structure preserving method adapted for local image editing using the "copy background" method from [6].

Approach	KW-mIoU (↑)	mIoU (↑)	FID (↓)	CLIP (↑)
Real Images	83.3	76.3	-	0.15
MS-COCO Shape				
Blended Diffusion [6]	23.3	41.8	46.2	0.20
SD-Inpaint [7]	38.5	51.7	43.7	0.19
SDEdit + Shape [8]	31.0	49.9	45.1	**0.21**
P2P + Shape [5]	46.9	63.3	**39.6**	0.20
Ours (w/o IOA)	43.8	55.3	41.5	0.21
Ours	**53.3**	**63.6**	40.2	**0.21**
Inferred Shape				
P2P [5]	24.2	64.6	97.5	**0.26**
P2P + Shape [5]	37.7	54.0	51.1	0.21
Ours (w/o IOA)	33.0	46.0	56.8	0.22
Ours	**43.0**	**54.9**	**49.5**	0.22

Table 5.2: Automatic evaluation on MS-COCO ShapePrompts (test set). MS-COCO Shape uses instance masks provided by MS-COCO, and Inferred Shape uses masks inferred from the text. Ours w/o IOA denotes our method without Inside-Outside Attention.

Baselines For our baselines, we compare against the local image editing method Blended Diffusion [6], the inpainting method SD-Inpaint [7], and the structure preserving methods SDEdit [8] and P2P [5]. Blended Diffusion, built on top of a Guided Diffusion [194] backbone, uses mask input by copying the source image's background at each timestep and text input by applying classifier guidance with CLIP [147]. SD-Inpaint, built on top of a Stable Diffusion [191] backbone, finetunes the model with an extra U-Net channel to use mask input and applies classifier-free guidance to use text input. SDEdit partially noises then denoises the source image and P2P copies cross attention maps to preserve structure, and they apply classifier-free guidance to use text input. For the structure preserving methods we use implementations built on top of a Stable Diffusion backbone, and in some experiments we adapt them to use mask input by applying the "copy background" method from [6].

Experimental Setup For all baselines we use the default hyperparameters provided by their respective repositories. For sampling we use a standard DDIM scheduler for 50 inversion and generation steps. When using Inside-Outside Attention on cross-attention layers, we evenly divide the maximum number of text tokens excluding the `<bos>` token, resulting in 38 "inside" tokens and 38 "outside" tokens. The attentions for the `<bos>` token are zeroed out.

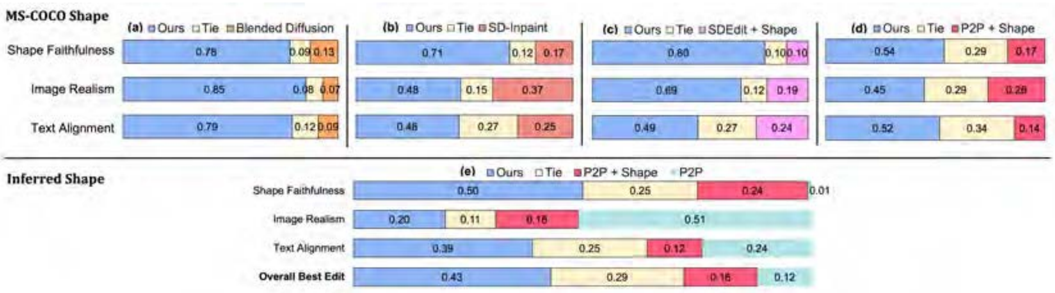


Figure 5.6: Annotator evaluation on MS-COCO ShapePrompts (100-sample subset of test set). Columns (a, b, c, d): we asked people to rate edits performed by our method vs. a baseline, where the two edits were presented as anonymized and in randomized order. Rows (shape faithfulness, image realism, text alignment): annotators selected the superior edit along these three axes. Each bar denotes the percentage of samples where the superior edit was "Ours", "Tie", or a baseline. In (e) we use the same procedure, except we presented three anonymized edits, ours vs. two baselines. Annotators were additionally asked to select the "overall best edit." We provide further details in the Supplemental.

5.5.1 Comparison to Prior Work

MS-COCO Shape We first experiment with instance-level masks provided by MS-COCO as our shape guidance. In Figure 5.5, we depict real images (first row) and edits made by Blended Diffusion (second row), SD-Inpaint (third row), and SDEdit + Shape (fourth row), P2P + Shape (fifth row), Ours (sixth row). Prior works demonstrate a variety of failure modes in shape-guided editing, where an object may be transformed into a new shape, removed completely, severely downscaled, or fail to respect the text prompt. On the other hand, our method is able to simultaneously respect the shape and the prompt without a compromise in image realism. As seen in Table 5.2, our method outperforms the local editing and inpainting baselines [6, 7] across the board, with at least a 15 point improvement in KW-mIoU. Comparing with the structure preserving baselines [5, 8], we achieve at least a 6 point improvement in KW-mIoU with comparable FID and CLIP scores. Intuitively, the baselines have trouble achieving shape faithfulness because "copy background" only provides shape signal based on how realistic the visual output looks at each timestep – the diffusion model attempts to rectify its edits based on how well it blends with the copied background. In Figure 5.7 we demonstrate this shape signal is weak in early diffusion timesteps where the output looks similar to pure noise, meaning that the model can irrecoverably produce an object with the incorrect scale or pose. Our Inside-Outside attention mechanism provides much stronger shape signal in early timesteps, where we enforce the location and scale of the object via the attention constraint.

We also conducted an evaluation with annotator ratings. We created four evaluations corresponding to each baseline, each of which contained 100 samples comparing an edit made by our method vs. the baseline in an anonymized and randomized fashion. For each sample, we asked five people to select the superior edit along the axes of shape faithfulness, image realism, and text alignment. As seen in the top row of Figure 5.6, annotators confirm that our method outperforms the baselines in shape faithfulness, with our method selected as superior at least 54% of the time (3.2x the most competitive baseline P2P + Shape). For image realism and text alignment, our method was selected as superior at least 48% of the time (1.3x and 1.9x the most competitive baseline SD-Inpaint).

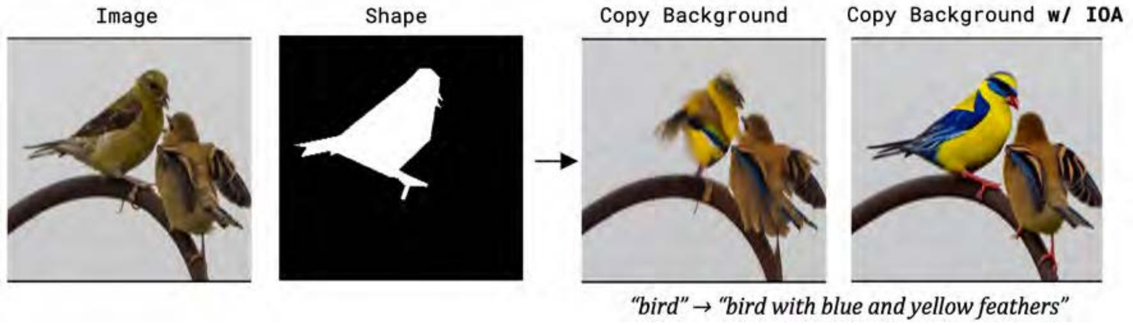


Figure 5.7: Shape signal from "copy background" is weak in early timesteps. In both examples we only use shape guidance in the first half of generation, where Inside-Outside Attention (w/ IOA) is able to provide stronger shape signal.

Inferred Shape Next, we demonstrate that our method also works on automatically inferred masks, which encompass a variety of challenging cases, such as reflections or multiple overlapping instances (Figure 5.1). We compare against our most competitive baseline, vanilla P2P and P2P adapted for local image editing using the inferred mask (P2P + Shape). P2P often produces edits that look nothing like the source image (see Figure 5.1), explaining how it has the worst FID scores in Table 5.2 (its distribution of synthetic images is significantly different than the real images) but the best image realism ratings in Figure 5.6. In a similar fashion, it achieves a better CLIP score but worse text alignment rating than our method because it is easier to align with the prompt if significantly deviating from the source image. As a result, P2P is rated as the worst overall image *editing* method, as seen in the bottom row of Figure 5.6. In contrast, our method is rated as the best edit for 43% of samples, 2.7x more than the most competitive baseline P2P + Shape.

Ablations In Table 5.2 we ablate our method without Inside-Outside Attention (Ours w/o IOA) and with Inside-Outside Attention (Ours). We demonstrate that the mechanism is a critical component of our method, providing a 9.5 point and 10 point increase in KW-mIoU in the MS-COCO Shape and Inferred Shape settings respectively. Ours w/o IOA performs better than all baselines on all metrics, except P2P + Shape (only P2P and our method use inversion), demonstrating how DDIM inversion is another critical component. In the Supplemental we also ablate the use of DDIM inversion, guidance scale hyperparameters, and the use of a soft vs. hard shape constraint on the self-attention maps.

5.5.2 Additional Editing Results

In Figure 5.8, we demonstrate additional capabilities of our method beyond object editing. (a) Our method is able to perform both intra- and inter- class edits on the same image, for example editing a cow to wear "gold and diamond chains" or transform into a "sheep." (b) Our method is able to perform outside edits, whether it is a background "at sunset" or "in front of the Eiffel Tower in Paris." Interestingly, our method sometimes maintains structures from the real image, for example transforming the cabinet into a landmass in both edited images. (c) Our method is able to perform simultaneous edits with one prompt for the inside region ("...robot horse..") and another for the outside region ("...Big Ben..." or "...Metropolitan Museum of Art..."). Since our method delineates edits on the object vs. background, although every pixel in the image is transformed we can maintain the object-background relation from the source scene (e.g., the horse grazing). In contrast, it is not obvious how to adapt structure preserving methods for this simultaneous editing setting, since with "copy background" they require one region (e.g., the background) to remain identical to the source image to enforce locality.

5.6 Conclusion

In this work, we present the usefulness of an explicit shape for local edits on real images. We show that prior work in local editing, structure preserving editing, and inpainting often fail to respect shape. To alleviate this issue, we propose Shape-Guided Diffusion, a training-free method that uses a novel Inside-Outside Attention mechanism during both the inversion and generation process, which localizes object vs. background edits. We evaluate our method on our newly proposed MS-COCO ShapePrompts benchmark on the shape-guided editing task, where the goal is to edit an object given an input mask and text prompt. We show that our method significantly outperforms the baselines in shape faithfulness without a degradation

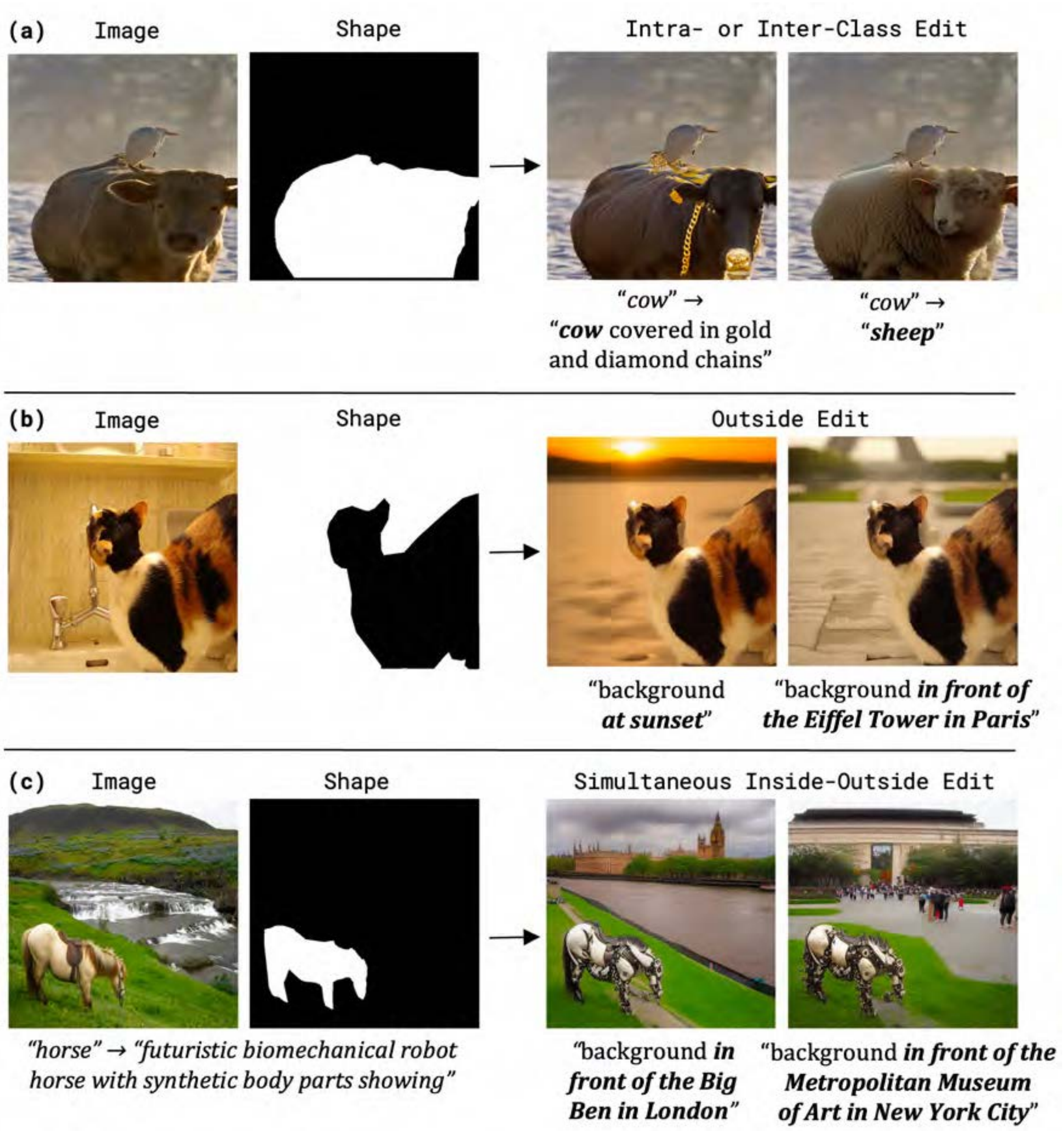

(a)
Image
Shape
Intra- or Inter-Class Edit
"cow" → "cow covered in gold and diamond chains"
"cow" → "sheep"
(b)
Image
Shape
Outside Edit
"background at sunset"
"background in front of the Eiffel Tower in Paris"
(c)
Image
Shape
Simultaneous Inside-Outside Edit
"horse" → "futuristic biomechanical robot horse with synthetic body parts showing"
"background in front of the Big Ben in London"
"background in front of the Metropolitan Museum of Art in New York City"

Figure 5.8: Additional editing results. Our method can perform intra- or inter-class edits on the same image, outside edits, and simultaneous inside-outside edits.

in text alignment or image realism when using either precise MS-COCO instance masks or masks inferred from the text.

Chapter 6

Summary and Future Directions

In this thesis, we explored various avenues from which multimodal generative models can be extended to provide a deeper understanding and enhanced performance in various tasks. Specifically, the study delves into image-to-text models, focusing on visual question answering (VQA), activity recognition, and image captioning, and into text-to-image models concerning image synthesis and shape-guided editing. Chapter 2 offers a method to enhance VQA systems' explainability, emphasizing the combined use of textual rationale and attention visualization. This approach was shown to improve model outputs and their ability to localize decision evidence. Chapter 3 diverges from traditional text generation orders, presenting an unsupervised learner that discovers optimal text generation sequences from training data. Using this method, improvements were observed in code generation, machine translation, and image captioning. Chapter 4 introduces a novel framework for semantic diffusion guidance, enabling the repurposing of unconditional image diffusion models to become text and/or image conditional without retraining. This framework is particularly advantageous for datasets lacking text annotations. Chapter 5 identifies and addresses a problem in existing text-to-image diffusion models for editing task where object shape must be preserved. By introducing a novel Inside-Outside Attention mechanism, the study ensures that modifications are accurately associated with the intended spatial region.

This thesis focuses on uni-directional generation tasks where a generation of one modality is conditioned on the other (i.e. either image-to-text or text-to-image). As much as these uni-directional objectives are useful for training multimodal models with great versatility, less exploration has been made on whether training under bi-directional objectives can be beneficial. Whether we can utilize limited resources more efficiently by having a single model that can simultaneously generate both types of modalities is an open question. Whether joint generation of texts and images can lead to more interesting generation patterns or task extension is also a research direction that is yet to be explored. This thesis delves into various architectures and optimization techniques, aiming to unlock novel generation capabilities for multi-modal applications. We hope the findings from this research can serve as a foundation for the development of more advanced vision and language models and provide a clearer roadmap for future investigations in the field of multi-modal generative modeling.

Bibliography

[1] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014. iv, 6, 8, 15, 16, 17

[2] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *CVPR*, 2014. iv, 5, 6, 8, 15, 16

[3] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in Neural Information Processing Systems 12* (S. A. Solla, T. K. Leen, and K. Müller, eds.), pp. 1057–1063, MIT Press, 2000. v, 21, 26

[4] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017. v, 21

[5] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-Or, "Prompt-to-prompt image editing with cross attention control," *arXiv preprint arXiv:2208.01626*, 2022. vii, viii, 53, 55, 63, 64, 65

[6] O. Avrahami, D. Lischinski, and O. Fried, "Blended diffusion for text-driven editing of natural images," in *CVPR*, 2022. viii, 55, 63, 64, 65

[7] R. ML, "Stable diffusion inpainting." https://huggingface.co/runwayml/stable-diffusion-inpainting, 2022. viii, 56, 63, 64, 65

[8] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, "SDEdit: Guided image synthesis and editing with stochastic differential equations," in *International Conference on Learning Representations*, 2022. viii, 55, 63, 64, 65

[9] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014. 1, 5, 10

[10] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014. 1

[11] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013. 1

[12] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014. 1, 22

[13] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, "Improving language understanding by generative pre-training," 2018. 1, 22

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017. 1, 39

[15] H. Xu, G. Ghosh, P.-Y. Huang, P. Arora, M. Aminzadeh, C. Feichtenhofer, F. Metze, and L. Zettlemoyer, "Vlm: Task-agnostic video-language model pre-training for video understanding," *arXiv preprint arXiv:2105.09996*, 2021. 1

[16] Z. Wang, J. Yu, A. W. Yu, Z. Dai, Y. Tsvetkov, and Y. Cao, "Simvlm: Simple visual language model pretraining with weak supervision," *arXiv preprint arXiv:2108.10904*, 2021. 1

[17] P. Wang, A. Yang, R. Men, J. Lin, S. Bai, Z. Li, J. Ma, C. Zhou, J. Zhou, and H. Yang, "Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework," in *International Conference on Machine Learning*, pp. 23318–23340, PMLR, 2022. 1

[18] M. Tsimpoukelli, J. L. Menick, S. Cabi, S. Eslami, O. Vinyals, and F. Hill, "Multimodal few-shot learning with frozen language models," *Advances in Neural Information Processing Systems*, vol. 34, pp. 200–212, 2021. 1

[19] J. Cho, J. Lei, H. Tan, and M. Bansal, "Unifying vision-and-language tasks via text generation," in *International Conference on Machine Learning*, pp. 1931–1942, PMLR, 2021. 1

[20] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, *et al.*, "Flamingo: a visual language model for few-shot learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 23716–23736, 2022. 1

[21] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36479–36494, 2022. 1

[22] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, "Glide: Towards photorealistic image generation and editing with text-guided diffusion models," *arXiv preprint arXiv:2112.10741*, 2021. 1, 39

[23] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, 2022. 1, 53

[24] O. Gafni, A. Polyak, O. Ashual, S. Sheynin, D. Parikh, and Y. Taigman, "Make-a-scene: Scene-based text-to-image generation with human priors," in *European Conference on Computer Vision*, pp. 89–106, Springer, 2022. 1, 56

[25] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022. 1

[26] J. Yu, Y. Xu, J. Y. Koh, T. Luong, G. Baid, Z. Wang, V. Vasudevan, A. Ku, Y. Yang, B. K. Ayan, *et al.*, "Scaling autoregressive models for content-rich text-to-image generation," *arXiv preprint arXiv:2206.10789*, vol. 2, no. 3, p. 5, 2022. 1

[27] H. Chang, H. Zhang, J. Barber, A. Maschinot, J. Lezama, L. Jiang, M.-H. Yang, K. Murphy, W. T. Freeman, M. Rubinstein, *et al.*, "Muse: Text-to-image generation via masked generative transformers," *arXiv preprint arXiv:2301.00704*, 2023. 1

[28] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the v in vqa matter: Elevating the role of image understanding in visual question

answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 4

[29] E. H. Shortliffe and B. G. Buchanan, "A model of inexact reasoning in medicine," *Mathematical biosciences*, vol. 23, no. 3, pp. 351–379, 1975. 5

[30] H. C. Lane, M. G. Core, M. Van Lent, S. Solomon, and D. Gomboc, "Explainable artificial intelligence for training and tutoring," tech. rep., DTIC Document, 2005. 5

[31] M. Van Lent, W. Fisher, and M. Mancuso, "An explainable artificial intelligence system for small-unit tactical behavior," in *NCAI*, 2004. 5

[32] M. G. Core, H. C. Lane, M. Van Lent, D. Gomboc, S. Solomon, and M. Rosenberg, "Building explainable artificial intelligence systems," in *Proceedings of the national conference on artificial intelligence*, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006. 5

[33] L.-A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell, "Generating visual explanations," in *ECCV*, 2016. 5

[34] T. Berg and P. N. Belhumeur, "How do you tell a blackbird from a crow?," in *ICCV*, 2013. 5

[35] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. Efros, "What makes paris look like paris?," *ACM Transactions on Graphics*, vol. 31, no. 4, 2012. 5

[36] V. Escorcia, J. C. Niebles, and B. Ghanem, "On the relationship between visual attributes and convolutional networks," in *CVPR*, 2015. 5, 6

[37] B. Zhou, D. Bau, A. Oliva, and A. Torralba, "Interpreting deep visual representations via network dissection," *arXiv preprint arXiv:1711.05611*, 2017. 5

[38] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene cnns," in *ICLR*, 2015. 5, 6

[39] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," *arXiv preprint arXiv:1704.03296*, 2017. 5

[40] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *See https://arxiv. org/abs/1610.02391 v3*, vol. 7, no. 8, 2016. 5

[41] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pp. 818–833, Springer, 2014. 5, 6

[42] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling, "Visualizing deep neural network decisions: Prediction difference analysis," *arXiv preprint arXiv:1702.04595*, 2017. 5

[43] L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell, "Generating visual explanations," in *ECCV*, 2016. 6, 12, 13

[44] A. Das, H. Agrawal, C. L. Zitnick, D. Parikh, and D. Batra, "Human attention in visual question answering: Do humans and deep networks look at the same regions?," *CoRR*, vol. abs/1606.03556, 2016. 6, 9, 13, 14

[45] M. Malinowski, M. Rohrbach, and M. Fritz, "Ask your neurons: A neural-based approach to answering questions about images," in *ICCV*, 2015. 6

[46] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *CVPR*, 2016. 6

[47] H. Xu and K. Saenko, "Ask, attend and answer: Exploring question-guided spatial attention for visual question answering," in *ECCV*, 2016. 6

[48] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei, "Visual7W: Grounded Question Answering in Images," in *CVPR*, 2016. 6

[49] K. Chen, J. Wang, L.-C. Chen, H. Gao, W. Xu, and R. Nevatia, "Abc-cnn: An attention based convolutional neural network for visual question answering," *arXiv:1511.05960*, 2015. 6

[50] C. Xiong, S. Merity, and R. Socher, "Dynamic memory networks for visual and textual question answering," in *ICML*, 2016. 6

[51] K. J. Shih, S. Singh, and D. Hoiem, "Where to look: Focus regions for visual question answering," in *CVPR*, 2016. 6

[52] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, "Multimodal compact bilinear pooling for visual question answering and visual grounding," 2016. 6, 10

[53] J. Kim, K. W. On, J. Kim, J. Ha, and B. Zhang, "Hadamard product for low-rank bilinear pooling," *CoRR*, vol. abs/1610.04325, 2016. 6, 7

[54] G. Gkioxari, R. Girshick, and J. Malik, "Contextual action recognition with r* cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1080–1088, 2015. 7

[55] A. Mallya and S. Lazebnik, "Learning models for actions and person-object interactions with transfer to question answering," in *ECCV*, 2016. 7

[56] L. Pishchulin, M. Andriluka, and B. Schiele, "Fine-grained activity recognition with holistic and pose based features," pp. 678–689, Springer, 2014. 7

[57] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "Vqa: Visual question answering," in *ICCV*, 2015. 7

[58] C. L. Zitnick, A. Agrawal, S. Antol, M. Mitchell, D. Batra, and D. Parikh, "Measuring machine intelligence through visual question answering," *CoRR*, vol. abs/1608.08716, 2016. 8

[59] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 8, 12

[60] S. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw," 2016. 9

[61] S. Bell, P. Upchurch, N. Snavely, and K. Bala, "Opensurfaces: A richly annotated catalog of surface appearance," in *SIGGRAPH*, 2013. 9

[62] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *ACL*, pp. 311–318, 2002. 12

[63] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, vol. 29, pp. 65–72, 2005. 12

[64] C.-Y. Lin, "Rouge: a package for automatic evaluation of summaries," in *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, 2004. 12

[65] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *CVPR*, pp. 4566–4575, 2015. 12

[66] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "Spice: Semantic propositional image caption evaluation," in *ECCV*, 2016. 12

[67] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases.," in *ICCV*, 1998. 12

[68] O. Pele and M. Werman, "Fast and robust earth mover's distances," in *2009 IEEE 12th International Conference on Computer Vision*, pp. 460–467, IEEE, September 2009. 13

[69] B. Uria, M. Côté, K. Gregor, I. Murray, and H. Larochelle, "Neural autoregressive distribution estimation," *J. Mach. Learn. Res.*, vol. 17, pp. 205:1–205:37, 2016. 20

[70] M. Germain, K. Gregor, I. Murray, and H. Larochelle, "Made: Masked autoencoder for distribution estimation," vol. 37 of *Proceedings of Machine Learning Research*, (Lille, France), pp. 881–889, PMLR, 07–09 Jul 2015. 20

[71] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL* (A. Moschitti, B. Pang, and W. Daelemans, eds.), pp. 1724–1734, ACL, 2014. 21

[72] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 3104–3112, 2014. 21

[73] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp. 3156–3164, IEEE Computer Society, 2015. 21

[74] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation

with visual attention," in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015* (F. R. Bach and D. M. Blei, eds.), vol. 37 of *JMLR Workshop and Conference Proceedings*, pp. 2048–2057, JMLR.org, 2015. 21

[75] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015* (L. Màrquez, C. Callison-Burch, J. Su, D. Pighin, and Y. Marton, eds.), pp. 1412–1421, The Association for Computational Linguistics, 2015. 21

[76] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015. 21

[77] A. van den Oord, N. Kalchbrenner, L. Espeholt, K. Kavukcuoglu, O. Vinyals, and A. Graves, "Conditional image generation with pixelcnn decoders," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain* (D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, eds.), pp. 4790–4798, 2016. 21

[78] S. Welleck, K. Brantley, H. D. III, and K. Cho, "Non-monotonic sequential text generation," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA* (K. Chaudhuri and R. Salakhutdinov, eds.), vol. 97 of *Proceedings of Machine Learning Research*, pp. 6716–6726, PMLR, 2019. 21

[79] M. Stern, W. Chan, J. Kiros, and J. Uszkoreit, "Insertion transformer: Flexible sequence generation via insertion operations," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA* (K. Chaudhuri and R. Salakhutdinov, eds.), vol. 97 of *Proceedings of Machine Learning Research*, pp. 5976–5985, PMLR, 2019. 21, 30

[80] O. Vinyals, S. Bengio, and M. Kudlur, "Order matters: Sequence to sequence for sets," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2016. 21

[81] J. Gū, H. S. Shavarani, and A. Sarkar, "Top-down tree structured decoding with syntactic connections for neural machine translation and parsing," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (Brussels, Belgium), pp. 401–413, Association for Computational Linguistics, Oct.-Nov. 2018. 21

[82] D. Alvarez-Melis and T. S. Jaakkola, "Tree-structured decoding with doubly-recurrent neural networks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, 2017. 21

[83] F. Stahlberg, "Neural machine translation: A review," *ArXiv*, vol. abs/1912.02047, 2019. 21

[84] Y. Oda, H. Fudaba, G. Neubig, H. Hata, S. Sakti, T. Toda, and S. Nakamura, "Learning to generate pseudo-code from source code using statistical machine translation," in *Proceedings of the 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, ASE '15, (Lincoln, Nebraska, USA), pp. 574–584, IEEE Computer Society, November 2015. 21, 29

[85] L. Ruis, M. Stern, J. Proskurnia, and W. Chan, "Insertion-deletion transformer," *CoRR*, vol. abs/2001.05540, 2020. 21

[86] X. Chen, N. Mishra, M. Rohaninejad, and P. Abbeel, "Pixelsnail: An improved autoregressive generative model," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018* (J. G. Dy and A. Krause, eds.), vol. 80 of *Proceedings of Machine Learning Research*, pp. 863–871, PMLR, 2018. 22

[87] I. Sutskever, J. Martens, and G. E. Hinton, "Generating text with recurrent neural networks," in *ICML*, 2011. 22

[88] T. Mikolov *et al.*, "Statistical language models based on neural networks," *Presentation at Google, Mountain View, 2nd April*, vol. 80, p. 26, 2012. 22

[89] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019. 22

[90] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020. 22

[91] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3128–3137, 2015. 22

[92] N. Ford, D. Duckworth, M. Norouzi, and G. E. Dahl, "The importance of generation order in language modeling," *arXiv preprint arXiv:1808.07910*, 2018. 22

[93] Q. Sun, S. Lee, and D. Batra, "Bidirectional beam search: Forward-backward inference in neural sequence models for fill-in-the-blank image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6961–6969, 2017. 24

[94] L. Zhou, J. Zhang, and C. Zong, "Synchronous bidirectional neural machine translation," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 91–105, 2019. 24

[95] S. Mehri and L. Sigal, "Middle-out decoding," in *Advances in Neural Information Processing Systems*, pp. 5518–5529, 2018. 24

[96] K. Yamada and K. Knight, "A syntax-based statistical translation model," in *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pp. 523–530, 2001. 24

[97] E. Charniak, K. Knight, and K. Yamada, "Syntax-based language models for statistical machine translation," in *Proceedings of MT Summit IX*, pp. 40–46, Citeseer, 2003. 24

[98] C. Dyer, A. Kuncoro, M. Ballesteros, and N. A. Smith, "Recurrent neural network grammars," *arXiv preprint arXiv:1602.07776*, 2016. 24

[99] R. Aharoni and Y. Goldberg, "Towards string-to-tree neural machine translation," *arXiv preprint arXiv:1704.04743*, 2017. 24

[100] X. Wang, H. Pham, P. Yin, and G. Neubig, "A tree-based decoder for neural machine translation," *arXiv preprint arXiv:1808.09374*, 2018. 24

[101] W. Chan, N. Kitaev, K. Guu, M. Stern, and J. Uszkoreit, "Kermit: Generative insertion-based modeling for sequences," 2019. 24

[102] S. Welleck, K. Brantley, H. Daumé III, and K. Cho, "Non-monotonic sequential text generation," *arXiv preprint arXiv:1902.02192*, 2019. 24, 31

[103] J. Gu, Q. Liu, and K. Cho, “Insertion-based decoding with automatically inferred generation order,” *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 661–676, 2019. 24, 28, 29, 30, 31

[104] D. Emelianenko, E. Voita, and P. Serdyukov, “Sequence modeling with unconstrained generation order,” in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada* (H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. B. Fox, and R. Garnett, eds.), pp. 7698–7709, 2019. 24

[105] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013. 24

[106] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, “beta-vae: Learning basic visual concepts with a constrained variational framework,” in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, 2017. 24

[107] G. Mena, D. Belanger, S. Linderman, and J. Snoek, “Learning latent permutations with gumbel-sinkhorn networks,” in *International Conference on Learning Representations*, 2018. 24, 27

[108] A. Grover, E. Wang, A. Zweig, and S. Ermon, “Stochastic optimization of sorting networks via continuous relaxations,” *ArXiv*, vol. abs/1903.08850, 2019. 24, 34

[109] S. Linderman, G. Mena, H. Cooper, L. Paninski, and J. Cunningham, “Reparameterizing the birkhoff polytope for variational permutation inference,” vol. 84 of *Proceedings of Machine Learning Research*, (Playa Blanca, Lanzarote, Canary Islands), pp. 1618–1627, PMLR, 09–11 Apr 2018. 24

[110] O. Vinyals, M. Fortunato, and N. Jaitly, “Pointer networks,” in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 2692–2700, 2015. 24, 25

[111] J. R. Munkres, “Algorithms for the Assignment and Transportation Problems,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, pp. 32–38, March 1957. 27

[112] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 5998–6008, Curran Associates, Inc., 2017. 29, 34

[113] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 2978–2988, Association for Computational Linguistics, July 2019. 29, 34

[114] W. Ling, P. Blunsom, E. Grefenstette, K. Hermann, T. Kociský, F. Wang, and A. Senior, "Latent predictor networks for code generation," *ArXiv*, vol. abs/1603.06744, 2016. 29

[115] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2015. 29

[116] D. Graff, J. Kong, K. Chen, and K. Maeda, "English gigaword," 2003. 29

[117] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015. 29

[118] L. Wu, X. Tan, D. He, F. Tian, T. Qin, J. Lai, and T.-Y. Liu, "Beyond error propagation in neural machine translation: Characteristics of language also matter," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (Brussels, Belgium), pp. 3602–3611, Association for Computational Linguistics, Oct.-Nov. 2018. 29

[119] N. Ford, D. Duckworth, M. Norouzi, and G. Dahl, "The importance of generation order in language modeling," *ArXiv*, vol. abs/1808.07910, 2018. 29

[120] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Berlin, Germany), pp. 1715–1725, Association for Computational Linguistics, Aug. 2016. 29

[121] J. Gu, C. Wang, and J. Zhao, “Levenshtein transformer,” in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, pp. 11181–11191, Curran Associates, Inc., 2019. 30

[122] Y. Kim and A. M. Rush, “Sequence-level knowledge distillation,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, (Austin, Texas), pp. 1317–1327, Association for Computational Linguistics, Nov. 2016. 30

[123] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, (Philadelphia, Pennsylvania, USA), pp. 311–318, Association for Computational Linguistics, July 2002. 31

[124] M. Denkowski and A. Lavie, “Meteor universal: Language specific translation evaluation for any target language,” in *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, 2014. 31

[125] C.-Y. Lin, “ROUGE: A package for automatic evaluation of summaries,” in *Text Summarization Branches Out*, (Barcelona, Spain), pp. 74–81, Association for Computational Linguistics, July 2004. 31

[126] R. Vedantam, C. L. Zitnick, and D. Parikh, “Cider: Consensus-based image description evaluation,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp. 4566–4575, IEEE Computer Society, 2015. 31

[127] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, “A study of translation edit rate with targeted human annotation,” in *In Proceedings of Association for Machine Translation in the Americas*, pp. 223–231, 2006. 31

[128] J. Gu, J. Bradbury, C. Xiong, V. O. K. Li, and R. Socher, “Non-autoregressive neural machine translation,” in *5th International Conference on Learning Representations*, 2018. 31, 34

[129] P. Shaw, J. Uszkoreit, and A. Vaswani, “Self-attention with relative position representations,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, (New Orleans, Louisiana), pp. 464–468, Association for Computational Linguistics, June 2018. 34

[130] R. L. Plackett, "The analysis of permutations.," pp. 193–202, 1975. 34

[131] R. D. Luce, *Individual Choice Behavior: A Theoretical analysis.* New York, NY, USA: Wiley, 1959. 34

[132] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *arXiv preprint arXiv:1812.04948*, 2018. 37

[133] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," *arXiv preprint arXiv:1809.11096*, 2018. 37

[134] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," *CVPR*, 2019. 37

[135] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," *arXiv preprint arXiv:2102.12092*, 2021. 37, 38, 39, 55

[136] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *arXiv preprint arXiv:2105.05233*, 2021. 37, 40, 42, 45

[137] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," *ICCV*, 2017. 38

[138] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "Attngan: Fine-grained text to image generation with attentional generative adversarial networks," *CVPR*, 2018. 38

[139] J. Zhu, Y. Shen, D. Zhao, and B. Zhou, "In-domain gan inversion for real image editing," in *European conference on computer vision*, pp. 592–608, Springer, 2020. 38

[140] A. Shoshan, N. Bhonker, I. Kviatkovsky, and G. Medioni, "Gan-control: Explicitly controllable gans," *arXiv preprint arXiv:2101.02477*, 2021. 38

[141] T. Xiao, J. Hong, and J. Ma, "Elegant: Exchanging latent encodings with gan for transferring multiple face attributes," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 168–184, 2018. 38

[142] H. Dhamo, A. Farshad, I. Laina, N. Navab, G. D. Hager, F. Tombari, and C. Rupprecht, "Semantic image manipulation using scene graphs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5213–5222, 2020. 38

[143] D. Bau, H. Strobelt, W. Peebles, J. Wulff, B. Zhou, J.-Y. Zhu, and A. Torralba, "Semantic photo manipulation with a generative image prior," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, p. 59, 2019. 38

[144] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 172–189, 2018. 38

[145] S. Azadi, M. Fisher, V. G. Kim, Z. Wang, E. Shechtman, and T. Darrell, "Multi-content gan for few-shot font style transfer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7564–7573, 2018. 38

[146] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *arXiv preprint arXiv:2006.11239*, 2020. 38, 40, 55, 61

[147] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," *arXiv preprint arXiv:2103.00020*, 2021. 38, 40, 41, 43, 45, 47, 55, 62, 64

[148] R. Gal, O. Patashnik, H. Maron, G. Chechik, and D. Cohen-Or, "Stylegan-nada: Clip-guided domain adaptation of image generators," *arXiv preprint arXiv:2108.00946*, 2021. 39, 40, 45

[149] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprint arXiv:1506.03365*, 2015. 39, 45

[150] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *International Conference on Machine Learning*, pp. 1060–1069, PMLR, 2016. 39

[151] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw," *Advances in neural information processing systems*, vol. 29, pp. 217–225, 2016. 39

[152] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014. 39

[153] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "Attngan: Fine-grained text to image generation with attentional generative adversarial

networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1316–1324, 2018. 39

[154] M. Zhu, P. Pan, W. Chen, and Y. Yang, "Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5802–5810, 2019. 39

[155] M. Tao, H. Tang, S. Wu, N. Sebe, X.-Y. Jing, F. Wu, and B. Bao, "Df-gan: Deep fusion generative adversarial networks for text-to-image synthesis," *arXiv preprint arXiv:2008.05865*, 2020. 39

[156] B. Li, X. Qi, T. Lukasiewicz, and P. H. Torr, "Controllable text-to-image generation," *arXiv preprint arXiv:1909.07083*, 2019. 39

[157] H. Dong, S. Yu, C. Wu, and Y. Guo, "Semantic image synthesis via adversarial learning," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5706–5714, 2017. 39

[158] S. Zhu, R. Urtasun, S. Fidler, D. Lin, and C. Change Loy, "Be your own prada: Fashion synthesis with structural coherence," in *Proceedings of the IEEE international conference on computer vision*, pp. 1680–1688, 2017. 39

[159] X. Mao, Y. Chen, Y. Li, T. Xiong, Y. He, and H. Xue, "Bilinear representation for language-based image editing using conditional generative adversarial networks," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2047–2051, IEEE, 2019. 39

[160] S. Nam, Y. Kim, and S. J. Kim, "Text-adaptive generative adversarial networks: Manipulating images with natural language," *arXiv preprint arXiv:1810.11919*, 2018. 39

[161] M. Günel, E. Erdem, and A. Erdem, "Language guided fashion image manipulation with feature-wise transformations," *arXiv preprint arXiv:1808.04000*, 2018. 39

[162] B. Li, X. Qi, T. Lukasiewicz, and P. H. Torr, "Manigan: Text-guided image manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7880–7889, 2020. 39

[163] J. Chen, Y. Shen, J. Gao, J. Liu, and X. Liu, "Language-based image editing with recurrent attentive models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8721–8729, 2018. 39

[164] K. Joseph, A. Pal, S. Rajanala, and V. N. Balasubramanian, "C4synth: Cross-caption cycle-consistent text-to-image synthesis," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 358–366, IEEE, 2019. 39

[165] T. Qiao, J. Zhang, D. Xu, and D. Tao, "Mirrorgan: Learning text-to-image generation by redescription," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1505–1514, 2019. 39

[166] A. Razavi, A. van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with vq-vae-2," in *Advances in neural information processing systems*, pp. 14866–14876, 2019. 39

[167] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2414–2423, 2016. 39

[168] C. Li and M. Wand, "Combining markov random fields and convolutional neural networks for image synthesis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2479–2486, 2016. 39

[169] S. Gu, C. Chen, J. Liao, and L. Yuan, "Arbitrary style transfer with deep feature reshuffle," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8222–8231, 2018. 39

[170] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann, and E. Shechtman, "Controlling perceptual factors in neural style transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3985–3993, 2017. 39

[171] N. Kolkin, J. Salavon, and G. Shakhnarovich, "Style transfer by relaxed optimal transport and self-similarity," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10051–10060, 2019. 39

[172] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1501–1510, 2017. 39

[173] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Universal style transfer via feature transforms," *arXiv preprint arXiv:1705.08086*, 2017. 39

[174] X. Li, S. Liu, J. Kautz, and M.-H. Yang, “Learning linear transformations for fast image and video style transfer,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3809–3817, 2019. 39

[175] T. Park, J.-Y. Zhu, O. Wang, J. Lu, E. Shechtman, A. A. Efros, and R. Zhang, “Swapping autoencoder for deep image manipulation,” *arXiv preprint arXiv:2007.00653*, 2020. 39

[176] R. Abdal, Y. Qin, and P. Wonka, “Image2stylegan: How to embed images into the stylegan latent space?,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4432–4441, 2019. 39

[177] J. Choi, S. Kim, Y. Jeong, Y. Gwon, and S. Yoon, “Ilvr: Conditioning method for denoising diffusion probabilistic models,” *arXiv preprint arXiv:2108.02938*, 2021. 39, 40, 45, 46, 47, 48, 51

[178] A. Casanova, M. Careil, J. Verbeek, M. Drozdzal, and A. Romero-Soriano, “Instance-conditioned gan,” *arXiv preprint arXiv:2109.05070*, 2021. 40

[179] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics,” in *International Conference on Machine Learning*, pp. 2256–2265, PMLR, 2015. 40, 54

[180] Y. Song and S. Ermon, “Generative modeling by estimating gradients of the data distribution,” *arXiv preprint arXiv:1907.05600*, 2019. 40

[181] Y. Song and S. Ermon, “Improved techniques for training score-based generative models,” *arXiv preprint arXiv:2006.09011*, 2020. 40

[182] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-based generative modeling through stochastic differential equations,” *arXiv preprint arXiv:2011.13456*, 2020. 40, 55

[183] C. Meng, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, “Sdedit: Image synthesis and editing with stochastic differential equations,” *arXiv preprint arXiv:2108.01073*, 2021. 40

[184] O. Avrahami, D. Lischinski, and O. Fried, “Blended diffusion for text-driven editing of natural images,” *arXiv preprint arXiv:2111.14818*, 2021. 40

[185] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski, "Styleclip: Text-driven manipulation of stylegan imagery," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2085–2094, 2021. 40, 55

[186] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*, pp. 694–711, Springer, 2016. 44

[187] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 46

[188] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019. 47, 55

[189] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems*, pp. 6626–6637, 2017. 48

[190] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018. 48

[191] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *CVPR*, 2022. 53, 55, 56, 57, 64

[192] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *arXiv preprint arXiv:2205.11487*, 2022. 53, 55, 56

[193] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *ICLR*, 2021. 55, 56, 60

[194] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis.," in *NeurIPS*, 2021. 55, 56, 64

[195] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022. 55

[196] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, “Glide: Towards photorealistic image generation and editing with text-guided diffusion models,” in *ICML*, 2022. 55, 56, 60

[197] G. Kim, T. Kwon, and J. C. Ye, “Diffusionclip: Text-guided diffusion models for robust image manipulation,” in *CVPR*, 2022. 55

[198] T. Brooks, A. Holynski, and A. A. Efros, “Instructpix2pix: Learning to follow image editing instructions,” *arXiv preprint arXiv:2211.09800*, November 2022. 55

[199] R. Mokady, A. Hertz, K. Aberman, Y. Pritch, and D. Cohen-Or, “Null-text inversion for editing real images using guided diffusion models,” *arXiv preprint arXiv:2211.09794*, 2022. 55, 56

[200] N. Tumanyan, M. Geyer, S. Bagon, and T. Dekel, “Plug-and-play diffusion features for text-driven image-to-image translation,” *arXiv preprint arXiv:2211.12572*, 2022. 55

[201] C. H. Wu and F. De la Torre, “Unifying diffusion models’ latent space, with applications to cyclediffusion and guidance,” *arXiv preprint arXiv:2210.05559*, 2022. 55

[202] D. Valevski, M. Kalman, Y. Matias, and Y. Leviathan, “Unitune: Text-driven image editing by fine tuning an image generation model on a single image,” *arXiv preprint arXiv:2210.09477*, 2022. 55, 60

[203] S. Iizuka, E. Simo-Serra, and H. Ishikawa, “Globally and locally consistent image completion,” *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–14, 2017. 56

[204] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, “Image inpainting for irregular holes using partial convolutions,” in *ECCV*, 2018. 56

[205] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, “Free-form image inpainting with gated convolution,” in *ICCV*, 2019. 56

[206] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, “Generative image inpainting with contextual attention,” in *CVPR*, 2018. 56

[207] S. Zhao, J. Cui, Y. Sheng, Y. Dong, X. Liang, E. I. Chang, and Y. Xu, “Large scale image completion via co-modulated generative adversarial networks,” 2021. 56

[208] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, "Repaint: Inpainting using denoising diffusion probabilistic models," in *CVPR*, 2022. 56

[209] Y. Zeng, Z. Lin, and V. M. Patel, "Shape-guided object inpainting," *arXiv preprint arXiv:2204.07845*, 2022. 56

[210] S. Wang, C. Saharia, C. Montgomery, J. Pont-Tuset, S. Noy, S. Pellegrini, Y. Onoe, S. Laszlo, D. J. Fleet, R. Soricut, J. Baldridge, M. Norouzi, P. Anderson, and W. Chan, "Imagen editor and editbench: Advancing and evaluating text-guided image inpainting," *arXiv preprint arXiv:2212.06909*, 2023. 56

[211] R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-Or, "An image is worth one word: Personalizing text-to-image generation using textual inversion," *arXiv preprint arXiv:2208.01618*, 2022. 60

[212] B. Cheng, A. G. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," in *NeurIPS*, 2021. 61, 62

[213] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," in *ECCV*, 2014. 62

[214] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky, "Resolution-robust large mask inpainting with fourier convolutions," *arXiv preprint arXiv:2109.07161*, 2021. 62

[215] H. Caesar, J. Uijlings, and V. Ferrari, "Coco-stuff: Thing and stuff classes in context," in *CVPR*, 2018. 62

[216] H. Yu, Y. Xu, J. Zhang, W. Zhao, Z. Guan, and D. Tao, "Ap-10k: A benchmark for animal pose estimation in the wild," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 62

[217] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016. 62

[218] G. Parmar, R. Zhang, and J.-Y. Zhu, "On aliased resizing and surprising subtleties in gan evaluation," in *CVPR*, 2022. 62