



Text-Based Semantic Image Editing

Guillaume Couairon

► To cite this version:

Guillaume Couairon. Text-Based Semantic Image Editing. Artificial Intelligence [cs.AI]. Sorbonne Université, 2023. English. NNT : 2023SORUS248 . tel-04145957v2

HAL Id: tel-04145957

<https://hal.science/tel-04145957v2>

Submitted on 10 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE DE DOCTORAT DE SORBONNE UNIVERSITÉ
Spécialité **Informatique**
École Doctorale Informatique, Télécommunications et Électronique (Paris)

Edition sémantique d'images à partir de requêtes textuelles

Text-Based Semantic Image Editing

Présentée par
Guillaume Couairon

Dirigée par
Pr. Matthieu Cord

Pour obtenir le grade de
DOCTEUR de SORBONNE UNIVERSITÉ

Présentée publiquement le 6 juillet 2023

Devant le jury composé de :

Jury:

Pr. Tinne TUYTELAARS <i>Professor, KU Leuven</i>	Rapportrice
Pr. Joost VAN DE WEIJER <i>Senior Scientist, Universitat Autònoma de Barcelona</i>	Rapporteur
Pr. Zeynep AKATA <i>Professor, University of Tübingen</i>	Examinatrice
Pr. Benoît FAVRE <i>Professor, Université Aix-Marseille, Polytech Marseille</i>	Examineur
Pr. Jakob VERBEEK <i>Research Scientist, Meta AI</i>	Encadrant de Thèse
Pr. Holger SCHWENK <i>Research Scientist, Meta AI</i>	Encadrant de Thèse
Pr. Matthieu CORD <i>Professor, Sorbonne Université</i>	Directeur de thèse

CONTENTS

CONTENTS	iii
ABSTRACT	i
RÉSUMÉ	iii
REMERCIEMENTS	v
NOTATIONS	vii
1 INTRODUCTION	1
1.1 Context	1
1.2 Image Editing	3
1.3 Contributions	6
2 BACKGROUND AND POSITIONING	9
2.1 Joint image/text understanding	9
2.2 Image generation	12
2.3 Image editing	17
2.4 Positioning	20
3 RETRIEVAL-BASED IMAGE EDITING WITH MULTIMODAL QUERIES	23
3.1 Introduction	23
3.2 The SIMAT database	25
3.3 Embedding Editing strategies	29
3.4 Experiments	31
3.5 Conclusion	37
4 IMAGE EDITING WITH INSTANCE-BASED OPTIMIZATION	41
4.1 Introduction	41
4.2 Related Work on Image Editing in latent spaces	42
4.3 FLEXIT algorithm	44
4.4 Experiments	47
4.5 Conclusion	62
5 IMAGE EDITING WITH DIFFUSION MODELS	65
5.1 Introduction	65
5.2 Related work on Editing with Diffusion Models	66
5.3 DIFFEDIT algorithm	67
5.4 Experiments	72
5.5 Conclusion	85
6 IMAGE SYNTHESIS FROM SEMANTIC SEGMENTATION MAPS	89
6.1 Introduction	89
6.2 Related work	91
6.3 ZESTGUIDE algorithm	93

6.4	Experiments	97
6.5	Conclusion	107
7	CONCLUSION	109
7.1	Summary of contributions	109
7.2	Perspective for future work	110
7.3	Societal Impact and Ethical challenges	111
	BIBLIOGRAPHY	113
A	APPENDIX	131
A.1	Publications	131
A.2	ImageNet transformations dataset	135
A.3	DIFFEDIT Experiments on filtered COCO dataset	139
A.4	Theoretical results for DIFFEDIT	140

ABSTRACT

The aim of this thesis is to propose algorithms for the task of Text-based Image Editing (TIE), which consists in editing digital images according to an instruction formulated in natural language. For instance, given an image of a dog, and the query "Change the *dog* into a *cat*", we want to produce a novel image where the dog has been replaced by a cat, keeping all other image aspects unchanged (animal color and pose, background). The north-star goal is to enable anyone to edit their images using only queries in natural language.

One specificity of text-based image editing is that there is practically no training data to train a supervised algorithm. In this thesis, we propose different solutions for editing images, based on the adaptation of large multimodal models trained on huge datasets.

We first study a simplified editing setup, named Retrieval-based image editing, which does not require to directly modify the input image. Instead, given the image and modification query, we search in a large database an image that corresponds to the requested edit. We leverage multimodal image/text alignment models trained on web-scale datasets (like CLIP) to perform such transformations without any examples. We also propose the SIMAT framework for evaluating retrieval-based image editing.

We then study how to directly modify the input image. We propose FLEXIT, a method which iteratively changes the input image until it satisfies an abstract "editing objective" defined in a multimodal embedding space. We introduce a variety of regularization terms to enforce realistic transformations.

Next, we focus on diffusion models, which are powerful generative models able to synthesize novel images conditioned on a wide variety of textual prompts. We demonstrate their versatility by proposing DIFFEDIT, an algorithm which adapts diffusion models for image editing without finetuning. We propose a zero-shot strategy for finding automatically where the initial image should be changed to satisfy the text transformation query.

Finally, we study a specific challenge useful in the context of image editing: how to synthesize a novel image by giving as constraint a spatial layout of objects with textual descriptions, a task which is known as Semantic Image Synthesis. We adopt the same strategy, consisting in adapting diffusion models to solve the task without any example. We propose the ZESTGUIDE algorithm, which leverages the spatio-semantic information encoded in the attention layers of diffusion models.

RÉSUMÉ

L'objectif de cette thèse est de proposer des algorithmes pour la tâche d'édition d'images basée sur le texte (TIE), qui consiste à éditer des images numériques selon une instruction formulée en langage naturel. Par exemple, étant donné une image d'un chien et la requête "Changez le *chien* en un *chat*", nous voulons produire une nouvelle image où le chien a été remplacé par un chat, en gardant tous les autres aspects de l'image inchangés (couleur et pose de l'animal, arrière-plan). L'objectif de l'étoile du nord est de permettre à tout un chacun de modifier ses images en utilisant uniquement des requêtes en langage naturel.

Une des spécificités de l'édition d'images basée sur du texte est qu'il n'y a pratiquement pas de données d'entraînement pour former un algorithme supervisé. Dans cette thèse, nous proposons différentes solutions pour l'édition d'images, basées sur l'adaptation de grands modèles multimodaux entraînés sur d'énormes ensembles de données.

Nous étudions tout d'abord une configuration d'édition simplifiée, appelée édition d'image basée sur la recherche, qui ne nécessite pas de modifier directement l'image d'entrée. Au lieu de cela, étant donné l'image et la requête de modification, nous recherchons dans une grande base de données une image qui correspond à la modification demandée. Nous nous appuyons sur des modèles multimodaux d'alignement image/texte entraînés sur des ensembles de données à l'échelle du web (comme CLIP) pour effectuer de telles transformations sans aucun exemple. Nous proposons également le cadre SIMAT pour évaluer l'édition d'images basée sur la recherche.

Nous étudions ensuite comment modifier directement l'image d'entrée. Nous proposons FLEXIT, une méthode qui modifie itérativement l'image d'entrée jusqu'à ce qu'elle satisfasse un "objectif d'édition" abstrait défini dans un espace d'intégration multimodal. Nous introduisons des termes de régularisation pour imposer des transformations réalistes.

Ensuite, nous nous concentrons sur les modèles de diffusion, qui sont des modèles génératifs puissants capables de synthétiser de nouvelles images conditionnées par une grande variété d'invites textuelles. Nous démontrons leur polyvalence en proposant DIFFEDIT, un algorithme qui adapte les modèles de diffusion pour l'édition d'images sans réglage fin. Nous proposons une stratégie "zero-shot" pour trouver automatiquement où l'image initiale doit être modifiée pour satisfaire la requête de transformation de texte.

Enfin, nous étudions un défi spécifique utile dans le contexte de l'édition d'images : comment synthétiser une nouvelle image en donnant comme contrainte une disposition spatiale d'objets avec des descriptions textuelles, une tâche qui est connue sous le nom de synthèse d'image sémantique. Nous adoptons la même stratégie, consistant à adapter les modèles de diffusion pour résoudre la tâche sans exemple. Nous proposons l'algorithme *ZESTGUIDE*, qui exploite l'information spatiosémantique encodée dans les couches d'attention des modèles de diffusion.

REMERCIEMENTS

Je voudrais remercier toutes les personnes qui m'ont accompagné durant ces trois dernières années et sans qui cette thèse n'aurait pas pu voir le jour. En premier lieu, merci à Matthieu Cord, mon directeur de thèse, pour son soutien indéfectible à travers les bons et les mauvais moments, pour avoir su me guider tout au long de la thèse par ses conseils avisés et encouragements, et pour la formidable ambiance d'équipe qu'il insuffle au groupe des *Chordettes*. Merci à Holger Schwenk, mon encadrant côté Meta, pour m'avoir fait confiance pendant le stage au FAIR et ensuite pour la concrétisation de la thèse CIFRE avec Matthieu sur ce sujet ambitieux à la croisée des expertises de chacun. Merci en particulier du soutien malgré des travaux de thèse qui ont finalement dérivé sur des sujets majoritairement orienté vision. Merci à Matthijs Douze pour son expertise et ses conseils au début de la thèse, d'avoir su proposer un cap malgré des moments de doute quant à la pertinence de la proposition initiale de thèse. Un grand merci aussi à Jakob Verbeek d'avoir rejoint le projet au moment critique de la génération d'image, ce qui a permis de concrétiser trois beaux travaux (constituants chacun un chapitre de la thèse); cette thèse lui doit beaucoup. Merci pour l'énergie que tu as donné dans ce projet, et pour ta rigueur intellectuelle qui nous a permis d'améliorer grandement les expériences et argumentations de nos papiers.

Je voudrais maintenant remercier les co-thésards à qui les travaux présentés dans cette thèse doivent beaucoup : Merci à Asya Grechka pour le projet FlexIT, pour tous tes travaux qui ont permis d'enrichir le papier et pour avoir présenté le papier à CVPR; merci à Marlene Careil, co-premier auteur pour le projet Zest-Guide, pour tout ce que tu as fait et en particulier pour avoir passé énormément de temps sur le papier la dernière semaine après avoir dû changer de modèle de diffusion au dernier moment.

Merci à mes autres collaborateurs, Stéphane Lathuillère pour ZestGuide, et en particulier un grand merci à mes collaborateurs du projet *Forest Monitoring*, Camille Couprie, Jaimie Tolan, Eric Yang, Ben Nosarzewski et Huy Vo. Merci à Camille pour son énergie et sa détermination dans ce beau projet. Merci à mes collaborateurs et amis de la team *Chordettes*, Mustafa Shukor, Corentin Dancette (the cream), Alexandre Ramé, Arthur Douillard avec qui je suis heureux d'avoir pu travaillé, et merci pour les bons moments passés ensemble au labo. Côté Facebook, merci à Pierre Fernandez, Paul-Ambroise Duquenne, Jean-Baptiste Gaya, Alaa El-Nouby pour les projets en commun et discussions scientifiques, qu'elles aient abouti à un papier ou non.

Merci à tous les doctorants de l'équipe MLIA et de FAIR Paris, en particulier la nouvelle génération de doctorants à FAIR qui a su redynamiser l'équipe après la covid ; merci aussi à Virginie Do pour nos nombreuses sessions musicales.

Cette thèse n'aurait pas pu voir le jour sans le soutien de ma compagne Cécile, que je remercie infiniment. Enfin, j'en profite pour souhaiter bonne chance à Paul Couairon qui prend la relève au sein de l'équipe MLIA avec Nicolas Thome, sur des sujets d'édition vidéo guidée par du texte.

NOTATIONS

Input image to be modified	I or I_0
Source Text in editing query, image caption	S
Target Text in the editing query	T or Q
Editing Mask or Inpainting Mask	M
CLIP image encoder	C_i
CLIP text encoder	C_t
VQGAN decoder	D
VQGAN encoder	E
Number of steps in diffusion process	\mathcal{T}
Diffusion model's noise estimator	ϵ_θ
Real Image or generated image used in diffusion	\mathbf{x}_0
Pourcentage of steps used for encoding images in DIFFEDIT	r
Image encoded at step in diffusion r	\mathbf{x}_r
DDIM Encoding function at step r	E_r
DDIM Decoding function at step r	D_r
SDEdit noisy encoder at step r	G_r
Binary mask identifying an object	\mathbf{S}_i

INTRODUCTION

We first give an overview of the general context of this thesis. We then present in more details the main task that we tackle, Text-based Image Editing (TIE), with its challenges. The last section is devoted to outlining this thesis' contributions and main publications.

1.1 Context

Artificial intelligence (AI) is a field of computer science, where the aim is to develop algorithms and systems that can perform tasks that normally require human intelligence, such as perception, reasoning, learning, and decision-making. AI algorithms have been successfully applied to a wide range of fields, and are notably used in healthcare (analyzing medical images and diagnose diseases, Nasser and Abu-Naser 2019), entertainment (playing strategic games like Chess and Go, Silver et al. 2018), finance (detecting credit card fraud, Awoyemi et al. 2017), self-driving cars (recognizing traffic patterns and navigating complex environments, Grigorescu et al. 2020), recommendation systems (personalize product recommendations for customers, Batmaz et al. 2019), manufacturing (optimizing production processes and predicting equipment failures, Carvalho et al. 2019), agriculture (optimizing crop yields, monitoring soil health, Van Klompenburg et al. 2020), weather forecasting (Sinha et al. 2022), energy production (plasma confinement for nuclear fusion, Degraeve et al. 2022), social networks (moderating online communities and detecting hate speech, Kiela et al. 2020), programming (code completion and code translation, Roziere et al. 2020), and predict protein folding (Jumper et al. 2021). AI usually involves training algorithms and models to learn patterns and insights from data, without being explicitly programmed.

Computer Vision (CV) is a subfield of AI that aims at enabling computers to understand, interpret and process digital images and videos. Applications of computer vision include image and video analysis, object recognition, face recognition, gesture recognition, autonomous vehicles, robotics, medical imaging, and many more. It has become an important technology in various industries, including

healthcare, automotive, entertainment, and security, among others. A major task in computer vision is image classification, where an algorithm should attribute a text label to a given image, among a finite set of classes. The ImageNet benchmark (J. Deng et al. 2009a) was established to compare quantitatively different classification algorithms, and neural networks brought a revolution in 2012 when they showed superior performance on this task compared to traditional approaches (Alex Krizhevsky et al. 2012). Neural networks are algorithms loosely inspired from the human brain, that can be trained to perform any task for which we have a dataset of examples, like labelled images in image classification. They are the basis for almost all modern computer vision systems.

Natural Language Processing (NLP) is another subfield of AI concerned with human language. The goal of NLP is to enable computers to understand, interpret, and generate human language, both written and spoken. NLP encompasses tasks such as text classification, sentiment analysis, language translation, information extraction, question answering, and chatbot development, among others. The applications of NLP are very diverse and can be found in various industries, including healthcare, finance, customer service, marketing, and education. For example, in healthcare, NLP is used to extract insights from medical records, identify and categorize medical entities and concepts, and monitor patient data. In customer service, NLP is used to develop chatbots and virtual assistants that can interact with customers, answer questions, and provide personalized assistance. The most known and capable chatbot is ChatGPT: released in 2023, it reached 100 millions users in only 2 months. The core technology is a Large Language Model (LLM), trained with the next token prediction task, where the aim is to find the next word given the beginning of the sentence. Most modern NLP systems are based on the transformers neural network architecture (Ashish Vaswani et al. 2017). Beyond language-only applications, some models have integrated visual understanding of the world inside those language models, like Flamingo (Alayrac et al. 2022) or GPT-4 (Bubeck et al. 2023). This allows to perform tasks that require multimodal image and text understanding, like answering a question about a visual scene.

Text-to-image generation is arguably one of the biggest AI breakthroughs of the last three years. It consists in generating an image based on a textual description of that image, written in natural language. The training data is obtained by scrapping the web to find text-image pairs, which usually consist in an image with an associated caption. Text-to-image generation models are typically trained on billions of images and are able to generate photo-realistic images for a wide range of text prompts, include rare objects and unusual scene compositions, like *a teddy bear swimming* or *a corgi in a sushi house* (see Figure 1.1). These models have



A blue jay standing on a large basket of rainbow macarons.

Teddy bears swimming at the Olympics 400 M Butterfly event.

A flooded art gallery displaying Monet Paintings, with robots going around using paddle boards.

A cute corgi lives in a house made out of sushi.

Figure 1.1. – A few examples of images generated with the text-to-image generation algorithm Imagen (Saharia et al. 2022b). Images taken from <https://imagen.research.google/>.

also shown artistic potential, being able to imitate an artist’s style (Crowson et al. 2022). In this thesis, we are interested in the field of Text-Based Image Editing, where such algorithms are expected to not only generate novel images, but also *modify* existing images.

1.2 Image Editing

Image editing refers to the process of manipulating digital images using software tools. This can involve adjusting the color balance, brightness and contrast, cropping or resizing the image, removing unwanted objects or backgrounds, and applying filters or effects to the image. Image editing is an essential tool for graphic designers and other creative professionals, and it requires a wide range of skills and technical knowledge of specialized software like Photoshop. AI-powered image editing can help automate the more tedious and time-consuming aspects of image editing, allowing professionals to focus on more creative aspects of their work. Using generative modelling approaches for image editing is especially useful for the following applications:

- *Image restoration*: removing noise, blur, or other distortions from images (Isola et al. 2017);
- *Image enhancement*: improving image quality by adjusting brightness, contrast, and color saturation (Shi et al. 2020);

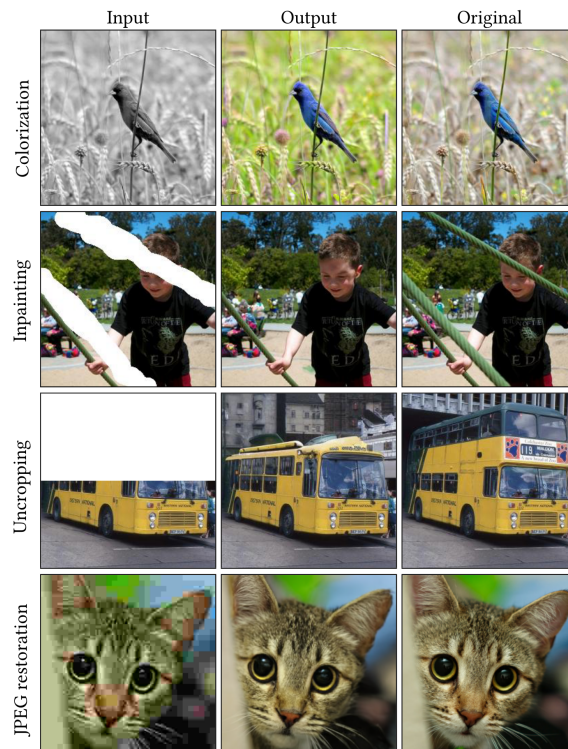


Figure 1.2. – Examples of tasks in image editing, with the "Output" middle column being the result of the Palette model (Saharia et al. 2022a) when given the input in the first column. Image from Saharia et al. (2022a).

- *Object Addition or Removal*: adding or removing objects or people from images (Gafni and Wolf 2020; Brown et al. 2022);
- *Image Inpainting* filling in missing areas of an image, such as scratches or damaged parts of a photo (Yu et al. 2018);
- *Style transfer*: applying the style of one image onto another image, such as turning a photo into a painting (Jing et al. 2019);
- *Super-Resolution*: increasing the resolution of images, making them appear sharper and more detailed (J. Johnson et al. 2016; Ledig et al. 2017; Saharia et al. 2022c);
- *Scene graph manipulation*: changing the interactions between scene elements (Dhamo et al. 2020);
- *Personalized image generation*: Placing subjects in novel contexts (Ruiz et al. 2022).

A few examples are shown in Figure 1.2.

In this thesis, we tackle the task of Text-based Image Editing (TIE), where the goal is to modify an input image based on an instruction written in natural language. We focus on high-level semantic modifications of the image scene, such as replacing objects or persons, or changing the interactions between the different scene elements. This requires computer vision algorithms able to understand and interpret the input image’s composition, as well as natural language processing algorithms able to understand the text instruction.

Text-based image editing has numerous applications. First, it can be used to analyze other AI-powered image processing algorithms. Computer vision models are often trained with the objective to classify images into a set of categories, like medical algorithms making disease predictions based on images or self-driving cars algorithms taking actions based on the visual perception of their environment. AI-powered image editing can be a powerful tool in understanding how these systems process their inputs, by studying how their outputs change when their inputs are changed. Second, text-based image editing is revolutionizing the fields of art and design, broadening the set of possibilities offered by text-to-image generation algorithms. Beyond text-to-image generation, iterative image editing can help visual creators to refine artistic ideas and incorporate parts of real photographs in their artwork. In the gaming industry, AI-based image generation and editing is used to create game environments, non-playable characters and other game assets. In e-commerce, text-based image editing has applications in helping users find clothing by iteratively refining a visual proposal based on a dialog.

From text-to-image generation to editing. Text-based image editing is closely related to text-to-image generation, because being able to synthesize novel objects is an essential skill to solve the task. Therefore, a central theme in this thesis is the question of how to leverage the knowledge acquired by text-to-image models trained on web-scaled data. This is crucial in TIE, because it is generally not possible to find supervised data (i.e. real editing examples) to train neural networks. While supervised data can be easily obtained on specific non-semantic image editing tasks like super-resolution and inpainting, relevant training data for semantic image editing requires either images edited by human experts, or pairs of very similar images annotated with a text describing the visual difference, both being very costly to acquire at scale. Zero-shot approaches are therefore the main approach for TIE, which consists in finding algorithms that solve the task by modifying text-to-image generation models, without any editing example. On top of this, training large text-to-image generation models has a very high computational cost.

1.3 Contributions

The aim of this thesis is to propose novel algorithms that leverage the power of large image and text processing models for image editing. We consider two classes of such models: dual image/text encoders trained to encode images and text data, which are not trained with an explicit image generation objective but learn robust multimodal representations; and diffusion models, which are models trained to generate images, currently representing the state-of-the-art in text-to-image generation.

This thesis is organized in five main chapters.

Chapter 2: Background and Positioning. In this chapter, we present the main papers that are at the foundation of our different works and contributions.

Chapter 3: Retrieval-based Image Editing with Multimodal queries. In this chapter, we study a simplified editing task that does not require image synthesis algorithms. Instead of editing the pixels of an image, we define a high-level specification of what the edited image should look, and search for a corresponding real image in a database. The material covered in this chapter was published in the following paper:

- Guillaume Couairon, Matthijs Douze, Matthieu Cord, and Holger Schwenk (2022a). “Embedding Arithmetic of Multimodal Queries for Image Retrieval”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, O-DRUM workshop*, pp. 4950–4958

Chapter 4: Image Editing with instance-based optimization of a multimodal embedding objective. In this chapter, we take advantage of the high-level image specification from the previous chapter, and we iteratively change the pixels of the input image to match this objective. The material covered in this chapter was published in the following paper:

- Guillaume Couairon, Asya Grechka, Jakob Verbeek, Holger Schwenk, and Matthieu Cord (2022b). “Flexit: Towards flexible semantic image translation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18270–18279

Chapter 5: Image Editing with diffusion models and mask guidance. In this chapter, we leverage synthesis abilities of *diffusion models*, a class of text-to-image generative models. We study how to adapt these models for image editing,

without any training. The material covered in this chapter was published in the following paper:

- Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord (2023b). “Diffedit: Diffusion-based semantic image editing with mask guidance”. In: *International Conference in Learning Representations*

Chapter 6: Image Synthesis conditioned on semantic segmentation maps. In this chapter, we consider a problem related to image editing: Semantic Image Synthesis, which consists in generating images conditioned on a spatial layout with spatial masks specifying in natural language what should appear in images. Tackling this problem is key in image editing, as it can be easily combined with inpainting methods to edit only parts of an image. The material covered in this chapter is currently under review in the following paper:

- Guillaume Couairon, Marlene Careil, Matthieu Cord, Stéphane Lathuillère, and Jakob Verbeek (2023a). “ZestGuide”. In: *Under Review*

The complete list of publications carried out during this thesis, involving collaborators from either Sorbonne Université or Meta AI, is presented in appendix [A.1](#).

BACKGROUND AND POSITIONING

In this chapter, we go over the main algorithms and methods at the foundation of our work, and recall the context in which these methods have been developed. We first present methods designed for joint understanding of vision and language, enabling us to solve multimodal tasks. Then, we give an overview of image generation algorithms, which are closely linked to our image editing objective. Finally, building on these generative models, we present a few baseline methods for image editing, that will be useful for understanding the next chapters. We finish by presenting our contributions, and positioning them in the wider context of image generation and editing.

2.1 Joint image/text understanding

Solving the task of text-based image editing, presented in the previous chapter, requires both natural language processing algorithms, to process the textual editing query, along with a detailed understanding of the visual world captured in digital images. It is not sufficient to have high-performing algorithms on each modality separately: editing algorithms need a correspondence between the visual appearance of digital images, and the natural language that we use to describe them.

From vision to vision-language The creation of the ImageNet dataset (Jia Deng et al. 2009) has paved the way for impressive advances in the design of neural networks architectures for computer vision (Alex Krizhevsky et al. 2012; He et al. 2015), mostly focused on the classification task. With 1000 fixed classes in the standard classification setup, neural networks learn to classify images without any meaningful understanding of the classes’ textual descriptions. These models are typically convolutional neural networks (CNNs) (LeCun, Bengio, et al. 1995; He et al. 2016), that apply convolution operations sequentially to the pixel representation of images, intertwined with non-linear operations and down-sampling operations. Beyond the vision-only approaches, understanding image and language jointly has been an important research direction of the machine learning

community since many years. It is indeed central to solving tasks involving interacting with the user in natural language, like Visual Question Answering (Goyal et al. 2017), where the goal is to answer a question about an image, written in natural language. First approaches (Karpathy and Fei-Fei 2015) involved using two pretrained neural networks: one for handling the question, pretrained on language-only data, and one for the image, typically pretrained on ImageNet. The network for language processing is usually a transformer (Ashish Vaswani et al. 2017), an architecture invented for sequence-to-sequence modelling that takes as input a list of tokens, each token representing a word or sub-word in the input text. These two networks were then combined in a single architecture with novel layers, fine-tuned to find the right answer among a fixed set of 3000 possible answers. More recent approaches, like OSCAR (X. Li et al. 2020) found that much better performance could be obtained by using a pretraining task that is inherently multimodal, learning from both images and text at the same time, which helps to understand the connection between vision and natural language. This requires parallel image-text data: in almost all cases, the pretraining database is a list of images described by textual captions. Among the most popular training objectives is masked language modelling (Devlin et al. 2018), introduced for Natural Language Processing, extended to multimodal pretraining. It consists in masking a fraction of words in the caption of an image-caption pair, and trying to recover the original words, from the image and unmasked caption words. Since solving this task requires high-level understanding of images, it provides a good objective for pre-training joint image/text models. To process image and text modalities jointly, the image is usually split into a set of token embeddings, by considering patches of pixels in the input image (Dosovitskiy et al. 2020; Touvron et al. 2021). The image token embeddings are then concatenated to the text token embeddings, before being processed by a single transformer, able to perform multimodal operations on this list of tokens coming from different modalities (W. Kim et al. 2021). This processing is called *Early fusion* as it fuses the different modalities together before processing it with the multimodal model.

CLIP Instead of early fusion, the other possibility for joint image/text understanding is to keep the multimodal pretraining objective, but instead of early fusion, to encode image and text modalities separately. This kind of architecture is called *dual encoders* (Liwei Wang et al. 2016; Faghri et al. 2018; Engilberge et al. 2018; Z. Zheng et al. 2020). Both images and text are encoded into a shared embedding space with disjoint networks, each specialized for their own modality: a transformer for text, and a convolutional network or vision transformer for the images. They are typically trained with the image-text matching (ITM) pretraining task on a database of captioned images (Sohn 2016b; Aaron van den Oord et al. 2018). The objective is to bring the image embeddings closer to the text embed-

dings of their respective captions, and farther from text embeddings of all other unrelated captions. Symmetrically, caption embeddings are brought closer to their associated images, and farther from unrelated images. This objective is illustrated in Figure 2.1. Authors of the CLIP algorithm (Radford et al. 2021a) have shown that this approach scales very well with dataset size: they use a large database of web-scraped 400M image-text pairs and reach competitive classification accuracy on ImageNet without using the training set, leveraging only the class names.

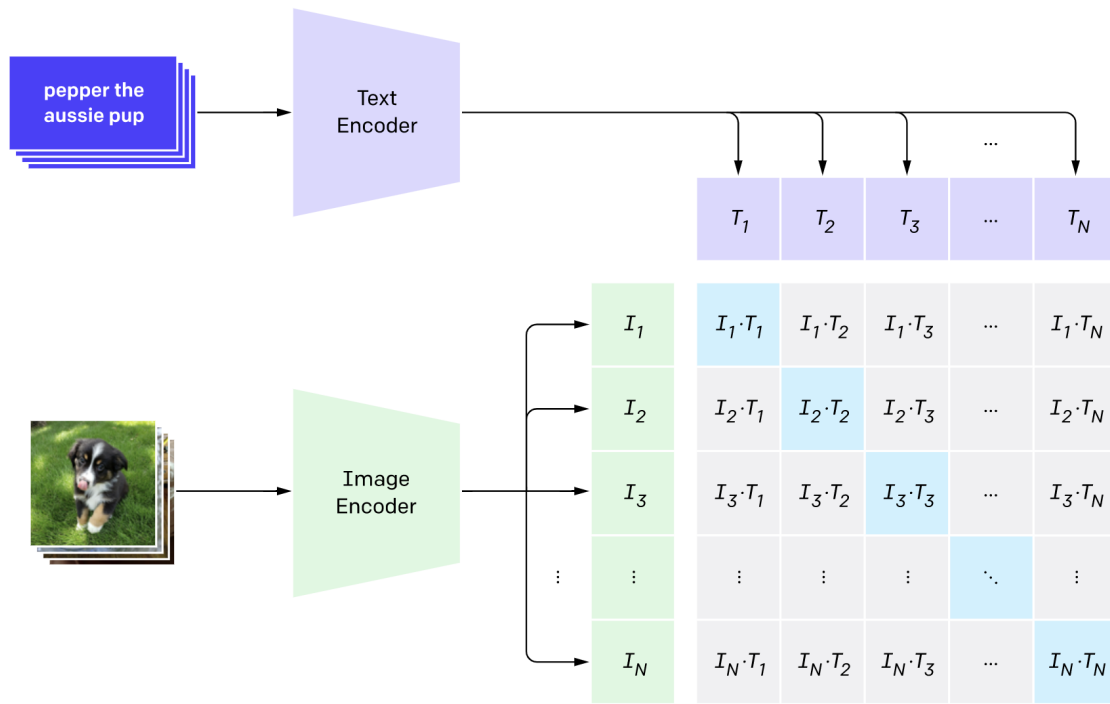


Figure 2.1. – Overview of the training objective used in CLIP. An image encoder and text encoder are trained to increase the similarity of matching image/text pairs (diagonal elements) while decreasing the similarity of mismatched image/text pairs (off-diagonal elements). Image from Radford et al. (2021a).

Subsequent works have shown that image-text matching is a very relevant pre-training objective for solving downstream multimodal tasks: using trained CLIP networks, both for image processing and language processing, provides a strong initialization scheme in most tasks requiring joint image/text understanding, such as image captioning (Barraco et al. 2022).

Once trained, CLIP gives a single representation space for both text and images, allowing to compute a normalized "alignment score" between an image and a caption, or even between two different images or two different texts. This representation space can be used for large-scale image-text retrieval.

2.2 Image generation

Text-based image editing requires algorithms trained for image synthesis. Image generation (or image synthesis) is the task of generating new images from a dataset. In probabilistic terms, it consists in learning to approximate the data probabilistic distribution p_{data} , such that the learned distribution p_{θ} can be sampled to produce novel images $x \sim p_{\theta}$. Conditional image generation consists in learning the conditional distribution $p_{\theta}(\cdot|c)$, where c is some additional information, which can be a class label, some text describing the image or another image in the case of image-to-image translation. In this section, we review a few different algorithms invented for image generation, and we introduce a few important tools that are heavily used in this thesis.

GANs Generative Adversarial Networks (GAN, Goodfellow et al. 2014) have long been the state-of-the art for generating images. It consists in co-training two networks, a generator and a discriminator: the discriminator learns to distinguish generated images from real images, while the generator learns to fool the discriminator. The generator typically takes Gaussian noise as input, so the generation process amounts to a single forward pass with a neural network, which is generally quite fast ($<0.1s$) for a single image. Text-conditional GANs can be trained by providing textual image descriptions as additional information to the generator and discriminator (Reed et al. 2016; Perarnau et al. 2016). However, GANs typically have two major issues: training stability and mode collapse, where parts of the training distribution are ignored and never generated by the generator. To solve these issues, other image generation algorithms have been proposed, which we review in the remainder of this section. However, GANs are still a very active area of research, and several recent work propose to fix the stability issue, e.g. by using pretrained discriminators (Sauer et al. 2022); and the mode collapse issue by using auxiliary losses, e.g. CLIP alignment for text-to-image generation (Kang et al. 2023).

Diffusion Models The current state-of-the art in image generation is dominated by Diffusion Models (Ho et al. 2020), that have taken over the field by storm, with Latent Diffusion Models (Rombach et al. 2022b), DALL-E 2 (Ramesh et al. 2022) and Imagen (Saharia et al. 2022b) vastly improving state of the art in modelling wide distributions of images and allowing for unprecedented compositionality of concepts in image generation.

The core idea is to train a model to reverse a diffusion process, which gradually adds Gaussian noise to an image I for a large number of time steps \mathcal{T} , until it is no longer distinguishable from random Gaussian noise. The diffusion process

gradually maps the data distribution to the unit Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. A neural network is then trained to reverse that process, mapping the unit Gaussian distribution to the data distribution. A novel image can be produced by sampling the unit Gaussian distribution, and iteratively denoising it with the trained neural network. It was shown that the training objective can equivalently be written as

$$\mathcal{L} = \mathbb{E}_{\mathbf{x}_0, t, \epsilon} \|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|_2^2, \quad (2.1)$$

where ϵ_θ is the noise estimator which aims to find the noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ that is mixed with an input image \mathbf{x}_0 to yield $\mathbf{x}_t = \sqrt{\alpha_t} \mathbf{x}_0 + \sqrt{1 - \alpha_t} \epsilon$. The coefficient α_t defines the level of noise and is a decreasing function of the time step t , with $\alpha_0 = 1$ (no noise) and $\alpha_T \approx 0$ (almost pure noise).

J. Song et al. (2021) propose to use ϵ_θ to generate new images with the Denoising Diffusion Implicit Model algorithm (DDIM): starting from $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the following update rule is applied iteratively until step 0:

$$\begin{aligned} \hat{\mathbf{x}}_0 &= \left(\frac{\mathbf{x}_t - \sqrt{1 - \alpha_t} \epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{\alpha_t}} \right) \\ \mathbf{x}_{t-1} &= \sqrt{\alpha_{t-1}} \hat{\mathbf{x}}_0 + \sqrt{1 - \alpha_{t-1}} \epsilon_\theta(\mathbf{x}_t, t). \end{aligned} \quad (2.2)$$

Text-conditional generation can be achieved by providing an encoding $\rho(T)$ of the text T as additional input to the noise estimator $\epsilon_\theta(\mathbf{x}_t, t, \rho(T))$ during training. The text is usually encoded with a frozen language-specific architecture, like CLIP in Stable Diffusion (Rombach et al. 2022b) or T5 (Raffel et al. 2020) for Imagen (Saharia et al. 2022b).

Auto-regressive sequence modelling Although images have a non-sequential spatial structure, it is possible to define a sequential representation for images and to learn the distribution with sequence modelling algorithms, borrowed from the NLP community. The most straightforward approach is to list the pixels composing an image in an arbitrary order, typically from left to right and top to bottom, giving a sequence of $N + 1$ real values (x_0, \dots, x_N) . Then, the sequence distribution is learned with the following decomposition: $p(x_0, \dots, x_N) = p(x_0) \prod_{i=1}^N p(x_i | x_1, \dots, x_{i-1})$, where a neural network is trained to predict the next pixel value x_i given the list of preceding pixel values x_0, \dots, x_{i-1} for every pixel index i from 0 to N . In this framework, learning text-conditional image generation $p(x_0, \dots, x_n, y)$ simply consists in prefixing the sequence of image pixels with the list of words in text y , for an adequate text representation. One of the most common architectures for sequence modelling is the transformer architecture (Ashish Vaswani et al. 2017), which has brought a revolution in NLP, and all modern large language models (LLM) are based on it. ImageGPT (M. Chen et al. 2020) has

demonstrated the high potential of this sequence modelling approach in image generation, yielding more stable and scalable results than GANs. This sequential modelling approach is also naturally suited for inpainting and image completion, closer to our image editing objective, since a fraction of the real pixel values in an input image can be provided to a model, that can fill in the rest with the learned model $p(x_i|x_0, \dots, x_{i-1})$. Despite the good stability and scalability results, this approach is very costly since it requires n forward passes in the transformer, one to generate each pixel value. As a result, generating images of size larger than 64x64 is too costly, especially given that transformers have an attention mechanism that scales quadratically with the sequence length.

Image Auto-Encoders Generating pixel values one by one is also very inefficient, since there is a lot of redundant information in images, especially locally where nearby pixels have highly correlated values. Therefore, better computational efficiency can be obtained by compressing images into shorter sequences. Efficient image compression was demonstrated with Vector Quantization approaches like VQ-VAE (A. van den Oord et al. 2017). In this line of work, Esser et al. (2021b) present an approach for image generation, with a two-stage image generation algorithm presented in Figure 2.2. The first stage consists in learning an encoder-decoder architecture (called VQGAN) to compress images into a sequence of tokens, where each token is a vector in a set called *codebook*, or *visual dictionary*. The token sequence length depends on the size of input image: typically, a 256x256 image is encoded in a 1024-long sequence. The second step is to model this compressed sequence distribution with a transformer trained with a next-token prediction objective. Once trained, the transformer is able to generate token sequences, which can be decoded into real images by using the image decoder trained during the first stage. Having this two-stage process allows to have a decoupling in image generation: the first stage specializes in compressing images with little information loss, while the transformer in the second-stage focuses on modelling higher-level non-trivial dependencies in images that are not captured by the first-stage model. One of the interesting properties of the first-stage encoder-decoder pair is that the decoder is trained to generate realistic textures, with a dedicated perceptual GAN objective comparing crops of the reconstructed image with crops of the input image being compressed. Therefore, although the compression algorithm does not maximize the pixel-wise reconstruction (which would give blurry results), it provides a more perceptual compression, where human can recognize that the texture in a reconstructed image is similar to the texture in the input image. This training procedure encourages the visual dictionary to encode more high-level perceptual information. Image auto-encoders like VQGAN have allowed for efficient auto-regressive modelling of images, notably with DALL-E (Ramesh et al. 2021), Cogview (Ding et al. 2021) and Make-a-scene

(Gafni et al. 2022a). The image encoder and decoders can be used without quantization in diffusion models: Latent Diffusion Models (Rombach et al. 2022a) are diffusion models trained in these image latent spaces instead of pixel spaces, yielding greater modelling efficiency: it indeed allows the diffusion model to focus on non-trivial image parts dependencies rather than low-level pixel correlations that are easily captured by the auto-encoder.

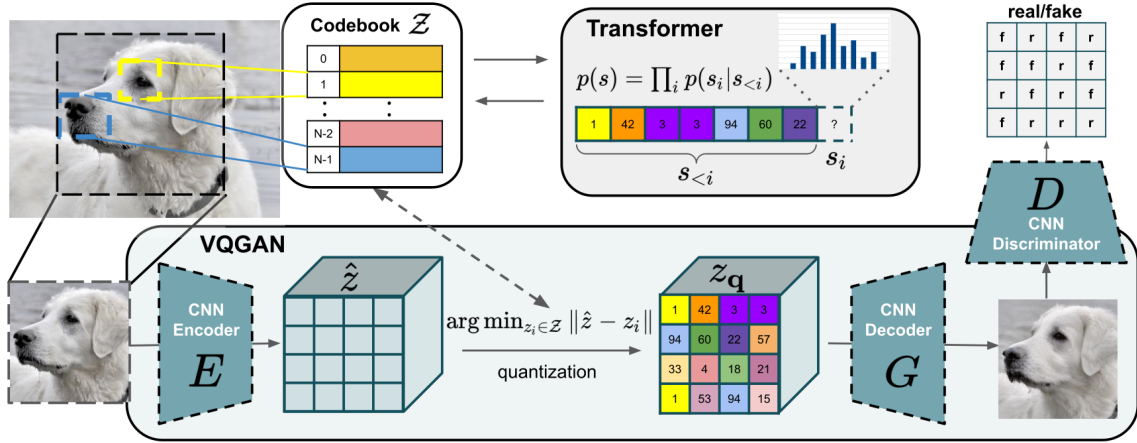


Figure 2.2. – Efficient auto-regressive image modelling with VQGAN. In a first stage, an encoder-decoder architecture is trained to compress images into a quantized latent space, effectively transforming images into sequences of tokens; In the second stage, a transformer is trained with auto-regressive modelling to generate token sequences. Image from Esser et al. (2021b).

VQGAN + CLIP The VQGAN encoder-decoder architecture provides a very good latent representation for images, that can be used for other tasks. We present here an algorithm that has sparked a revolution in the field of AI-based art generation, known as VQGAN+CLIP (Crowson et al. 2022), which combines this VQGAN latent representation with the CLIP multimodal encoders presented in Section 2.1. The VQGAN + CLIP text-to-image generation algorithm is based on the following idea: in the multimodal CLIP embedding space, we can compute the cosine similarity between the embeddings of an image I and a text description T , to measure whether T actually describes image I . This similarity score provides a differentiable objective that can be optimized by gradient descent on the image itself. Starting from a target text description T , a random initial image I is progressively optimized to increase CLIP similarity score with T . In a few hundred gradient descent steps, this produces an image well aligned in the CLIP embedding space, that should therefore be well described by input text T . By direct optimization of the pixel values composing the image, one quickly run into a problem related to adversarial attacks: the CLIP image backbone is

very confident that the resulting image matches the text description embedding (because we directly optimize the CLIP score), but the image does not look like a real image: finely adjusting the pixel values tricks the CLIP image backbone, but not the human eye. The problem is that there are too many dimensions in the pixel space, thus too many low-level parameters to optimize that, considered independently, do not reflect the actual variations of the visual world captured in images. The VQGAN latent space, on the other hand, compresses images into higher-level representations: it has a much lower vector space dimensionality and better reflects natural variations of image parts, at least locally. The VQGAN-CLIP algorithm consists in optimizing these latent codes directly, which means that the codes need to be decoded into real images before being processed by CLIP, which can only process RGB images. In this framework, the CLIP optimization problem is much more constrained: the optimized image stays in the space of images that consist of natural image parts, lessening the adversarial attack effect, and producing images that look more natural and match the text prompt. Generative artists have used VQGAN-CLIP to produce novel art images, quickly produce surprising and innovating image compositions (e.g. "portrait of Henri the 8th, cyberpunk edition") or imitate the style of known artists. Despite this success in art applications, the method is mostly adapted to art generation: the image generation objective is an implicit, inference-time optimization, which often produces non-photorealistic images.

2.3 Image editing

In this section, we start by presenting in detail our main task, text-based image editing. We then give an overview of a few baseline editing methods, which are foundational elements in our contributions.

Task setup and challenges. While Text-Based Image Editing is close to text-to-image generation, there are some specific challenges. First, while text-to-image algorithms are free to generate images with an unconstrained structure, having as sole requirement to match the text prompt, editing models should be able to adapt to the spatial structure of the input image. Second, editing performance cannot be measured with a single score, like in image classification. There are three requirements that edited images should meet:

- **Faithfulness:** the editing should be in accordance with the text editing query.
- **Edit distance:** the edited image should be as similar as possible to the input image: only parts of the image concerned by the editing should be changed, and all other image aspects should stay the same.
- **Image quality:** the edited image should look as natural as possible, ideally indistinguishable from a real image.

Evaluation. Each of these criteria can be measured with a dedicated metric. *Faithfulness* is measured by comparing the edited image with the text transformation query with a CLIP score (Radford et al. 2021a); *Edit distance* is measured with the LPIPS distance (R. Zhang et al. 2018), which is a perceptual distance between images; *Image quality* is measured with the FID score, a standard image quality assessment metric, which compares the distribution of generated images to the distribution of real images in the embedding space of an Inception network (Christian Szegedy et al. 2016).

It is not possible to optimize these three metrics at the same time. Matching the editing query and minimizing distance to input image are two contradictory objectives: there is an inherent trade-off between those two metrics. For a given editing method, better matching the text query comes at the cost of increased distance to the input image. Often, a parameter controls that trade-off, allowing generating Pareto curves between the metrics. Therefore, to compare different methods, we must compare their trade-off curves.

Image editing with GANs A lot of approaches for image editing have been using GANs. Some approaches involve training an end-to-end architecture with a proxy objective before being adapted to editing at inference time, based on GANs (B. Li et al. 2020b; B. Li et al. 2020a; Ma et al. 2018; Alami Mejjati et al. 2018; Mo et al. 2018; Gonzalez-Garcia et al. 2018). Other approaches have studied how to leverage a GAN pretrained on image generation: it has been explored to find directions in the latent space that correspond to specific semantic edits (Härkönen et al. 2020; Collins et al. 2020; Shen et al. 2020a; Shoshan et al. 2021), or to guide a latent space walk with an optimization objective in a multimodal embedding space like CLIP (Patashnik et al. 2021). These methods require GAN inversion to edit real images (T. Wang et al. 2022b; J. Zhu et al. 2020; Grechka et al. 2021b). We mention these approaches for reference, but we do not use GANs for image editing in this thesis.

Image Editing in embedding spaces In Section 2.1, we have presented the CLIP shared image-text embedding space, where images and text can be embedded into the same space with encoders C_i and C_t . In this space, the scalar product between image and text embeddings defines a similarity score, which is the basis for retrieval-based tasks. The image embeddings are compressed image representations, that encode the semantics in images rather than low-level pixel information, since they are designed to be compared with text embeddings. This semantic organization of the embedding space allows arithmetic manipulation of embedding vectors: similarly to geometric relationships in word embeddings (*King is to Queen what Man is to Woman*), semantic transformations (e.g. change *man* to *woman*) can be defined as vector directions (e.g., $C_t(woman) - C_t(man)$), and then added to image embeddings $C_i(I)$. More generally, to apply the text-defined transformation *change* T_1 into T_2 to an image embedding $C_i(I)$, we use the following simple equation:

$$P = C_i(I) + C_t(T_2) - C_t(T_1), \quad (2.3)$$

in which P should be the embedding of the input image I edited as requested by the text transformation query. In the case of the *man*→*woman* transformation, the gender of the main person depicted in the image should change accordingly. Now, the transformation only happens in the embedding space. To produce an image corresponding to the edited embedding P , we can either train an image generative model conditioned on CLIP embeddings, or we can simply retrieve in a large database, the image whose embedding is closest to P .

Image editing with instance-based optimization As explained above, the CLIP space can be used to perform editing on embeddings with simple linear opera-

tions. This embedding editing method can be coupled with a generative model like StyleGAN (T. Karras et al. 2020) to make real modifications on the input image (Patashnik et al. 2021). Instead of using a trained generative model, images can be synthesized by iterative modification of a random latent code from an image auto-encoder like VQGAN 2.2, to optimize the CLIP similarity score with a given text description. This generative algorithm can be easily extended to editing an input image I in two ways: first, instead of starting from a random latent code, we can start from the one representing the input image I , using simply the VQGAN encoder. The parameters of the iterative optimization procedure (longer training, larger learning rate) allow controlling how far we can go from I . Second, instead of optimizing the CLIP similarity with a single text, we can define a high-level specification as a multimodal CLIP embedding and optimize the edited image to match this embedding in the CLIP space. The optimization procedure aims at solving the following problem:

$$z = \arg \max_z C_i(D(z)) \cdot (C_i(I) + C_t(T_2) - C_t(T_1)) \quad (2.4)$$

where D is the image decoder, and z is a latent code initialized as $C_i(I)$ (the latent code of image I).

Image editing with diffusion models The generation algorithm for diffusion models is remarkably well suited for image editing. Indeed, image editing can be viewed as image generation with additional constraints, given by the input image; and there are multiple ways to guide the generative process with inference-time constraints, without retraining the diffusion model. We present here three important methods used in this thesis.

First, the SDEdit algorithm (Meng et al. 2022) allows to process an input image, and to generate a novel image with a pretrained diffusion model, that is close to the input image. The distance to the input image is controlled by a parameter interpolating between copying the input image and unconstrained generation. The algorithm can be summarized as following: given an input image $I = x_0$, first add random Gaussian noise to it, $x_r = \sqrt{\alpha_r} x_0 + \sqrt{1 - \alpha_r} \epsilon$. Then, run the diffusion model starting from step r with Equation 5.2. The parameter r controls the level of editing: with the maximum value $r = T$, $\alpha_T \approx 0$ so x_r is almost equal to random Gaussian noise, which corresponds to the classical DDIM generation algorithm. By choosing an appropriate noise level and conditioning text in the generation phase, one can get a generated image that respects a text specification, while controlling the distance with the input image.

Second, the DDIM update rule (J. Song et al. 2021) is well suited for inpainting: in this update rule at time step t , the variable \hat{x}_0 can be seen as an estimation of

the final generated image. At the beginning, this estimation is very coarse and blurry, but throughout the generation process, it becomes more and more refined until it actually corresponds to the generated image. This estimation is used to compute the next denoised sample \mathbf{x}_{t-1} , which is a very interesting behavior in the context of image editing. Changing the value of $\hat{\mathbf{x}}_0$ between each denoising step allows to guide the generative process in various ways. Image inpainting is a form of constrained image generation where outside an inpainting mask M , the pixels of the generated image should be equal to the pixel values of input image I . Therefore, to guide the diffusion generative process, we can simply "copy-paste" the ground truth values from input image, outside the mask:

$$\tilde{\mathbf{x}}_0 = \hat{\mathbf{x}}_0 * M + (1 - M) * I \quad (2.5)$$

and the DDIM update rule now uses $\tilde{\mathbf{x}}_0$ instead of \mathbf{x}_0 :

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}} \tilde{\mathbf{x}}_0 + \sqrt{1 - \alpha_{t-1}} \epsilon_\theta(\mathbf{x}_t, t). \quad (2.6)$$

Third, the iterative nature of the decoding process in diffusion models allows so-called "guidance" techniques, such as *classifier guidance* (Sohl-Dickstein et al. 2015; Yang Song et al. 2021; Dhariwal and Nichol 2021a). This technique consists in guiding the decoding process with the gradient of a trained classifier, with the aim to generate images that belonging to a certain class according to this classifier. Dhariwal and Nichol (2021a) show that DDIM sampling can be extended to sample the posterior distribution $p(\mathbf{x}_0|c)$, where c is the class detected by the classifier, with the following modification for the noise estimator ϵ_θ :

$$\tilde{\epsilon}_\theta(\mathbf{x}_t) = \epsilon_\theta(\mathbf{x}_t) - \sqrt{1 - \alpha_t} \nabla_{\mathbf{x}_t} p(c|\mathbf{x}_t) \quad (2.7)$$

where $p(c|\mathbf{x}_t)$ is the classifier probability that the input \mathbf{x}_t belong to class c . The classifier can be trained to use noisy inputs \mathbf{x}_t , but we can also use an off-the-shelf classifier by using the denoised estimation $\tilde{\mathbf{x}}_0$ as input at each step. This technique has since been extended to take as input other constraints such as for the tasks of inpainting, colorization, and super-resolution (Saharia et al. 2022a).

2.4 Positioning

In this section, we present our different contributions, what problems they aim at solving, and what methods and tools they are built upon. The thesis is organized in four chapters:

Retrieval-based Image Editing with Multimodal queries. In Chapter 3, we study a simplified editing task, which does not require an image generation algorithm. Given an input image and a text transformation query written as a pair (*source*, *target*) (e.g. *cat* \rightarrow *dog*), the aim is to find an image in a large image database that corresponds to the input image, modified according to the text query. Since searching in a database will give us real images, there might not be an exact match for our transformation, so we need to have some flexibility in accepting database matches. Solving this task requires to define a score measuring how well an image matches the theoretically best edited image, and then select the image in the database according to that score. We show that multimodal embedding spaces like CLIP (presented in Section 2.1) allow performing arithmetic operations between image embeddings and text embeddings, allowing to add or remove semantic concepts with simple additions and subtractions. We express the editing constraint as a target image embedding for the edited image, and then search for the image that has the nearest embedding in the database. We consider the problem of image editing in a multimodal embedding space, which does not require image synthesis abilities. This chapter’s material is the foundation for the following publication:

- Guillaume Couairon, Matthijs Douze, Matthieu Cord, and Holger Schwenk (2022a). “Embedding Arithmetic of Multimodal Queries for Image Retrieval”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, O-DRUM workshop*, pp. 4950–4958

Image Editing with instance-based optimization. In Chapter 4, we are interested in real image editing, which modifies the pixels of the input image instead of performing database retrieval. The problematic is as following: how can we use the CLIP multimodal embedding space to perform real image editing, given that it has not been trained with an image generation objective? Inspired by the VQGAN+CLIP approach presented in Section 2.2, we perform editing as instance optimization of the input image’s VQGAN representation, with the aim to reach a target embedding in the CLIP space. We then propose a variety of regularization schemes to improve editing accuracy and realism. This chapter’s material is the foundation for the following publication:

- Guillaume Couairon, Asya Grechka, Jakob Verbeek, Holger Schwenk, and Matthieu Cord (2022b). “Flexit: Towards flexible semantic image translation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18270–18279

Image Editing with diffusion models. In Chapter 5, we leverage recent advances in generative models based on diffusion. Since these models are able to synthesize a wide range of visual objects and concepts, we study how to leverage these abilities in the context of image editing. The main challenge is to understand the image given as input, find places where the image need editing, generate the edit and then blend it seamlessly with the input image. We have three main contributions: (i) we design a method to find what should be changed in an image, given an input and editing query; (ii) we adapt DDIM encoding to better preserve information of the input image, and give theoretical insights; (iii) we provide a quantitative evaluation framework based on three datasets, focusing on different aspects of text-based image editing. This chapter’s material is the foundation for the following publication:

- Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord (2023b). “Diffedit: Diffusion-based semantic image editing with mask guidance”. In: *International Conference in Learning Representations*

Image Synthesis from semantic segmentation maps. In Chapter 6 we consider a problem related to image editing: Semantic Image Synthesis, which consists in generating images conditioned on a spatial layout with spatial masks specifying in natural language what should appear in images. We choose to design a zero-shot algorithm for this task by adapting pre-trained diffusion models on this task, without any supervised data. Similarly to image editing, adapting diffusion models to this task requires to incorporate an additional constraint into the generative algorithm, which is challenging since spatial layouts are strong constraints, which the diffusion models have not been trained to process. Our algorithm, dubbed ZESTGUIDE, is based on the classifier guidance method; we leverage the spatio-semantic patterns in the attention maps of the diffusion model’s U-Net to guide the generative algorithm. This chapter’s material is the foundation for the following submission, currently under review:

- Guillaume Couairon, Marlene Careil, Matthieu Cord, Stéphane Lathuillère, and Jakob Verbeek (2023a). “ZestGuide”. In: *Under Review*

RETRIEVAL-BASED IMAGE EDITING WITH MULTIMODAL QUERIES

3.1 Introduction

In Chapter 2, we have presented how images and text can be embedded in a shared multimodal latent space, which enables essential multimodal applications like large-scale image/text retrieval. Classical image/text retrieval consists in having a query from one modality, like a textual image description, and to find a matching sample from the other modality. In this chapter, we study the problem of image retrieval from *multimodal queries* (called Multimodal Image Retrieval): given an input image and a simple text transformation query (e.g. *cat*→*dog*), the aim is to find in a large database, an image for which the semantic difference with the input image (e.g. representing a *cat*) can be accurately described by the text transformation. For example, with the *cat*→*dog* transformation, an image showing a cat sitting in the grass should be transformed into an image with a dog sitting in the grass (Figure 3.1): the main semantic difference is accurately described by the transformation *cat*→*dog*.

Multimodal Image Retrieval is a simplified editing problem which focuses on semantic composition of visual concepts rather than image synthesis skills. Most existing methods for solving this task (N. Vo et al. 2019b; Anwaar et al. 2021; Yale Song and Soleymani 2019) focus on supervised learning, using a fraction of the dataset for training and the remaining for testing. Instead, we want to measure if multimodal embeddings trained with an image/text matching objective can be used to solve this task without any transformation example.

We choose to transform images by encoding the transformation query as a *delta vector* in the multimodal space, before adding it to an image embedding and retrieving the closest image in a database (see Figure 3.1). This operation solely relies on the image/text alignment without needing any transformation example. However, it requires a well-structured multimodal space to be able to transfer text transformations to images. We know that word and sentence embeddings trained on vast amounts of data have been shown to possess geometric properties that can be useful for text transformation (Mikolov et al. 2013; Logeswaran and

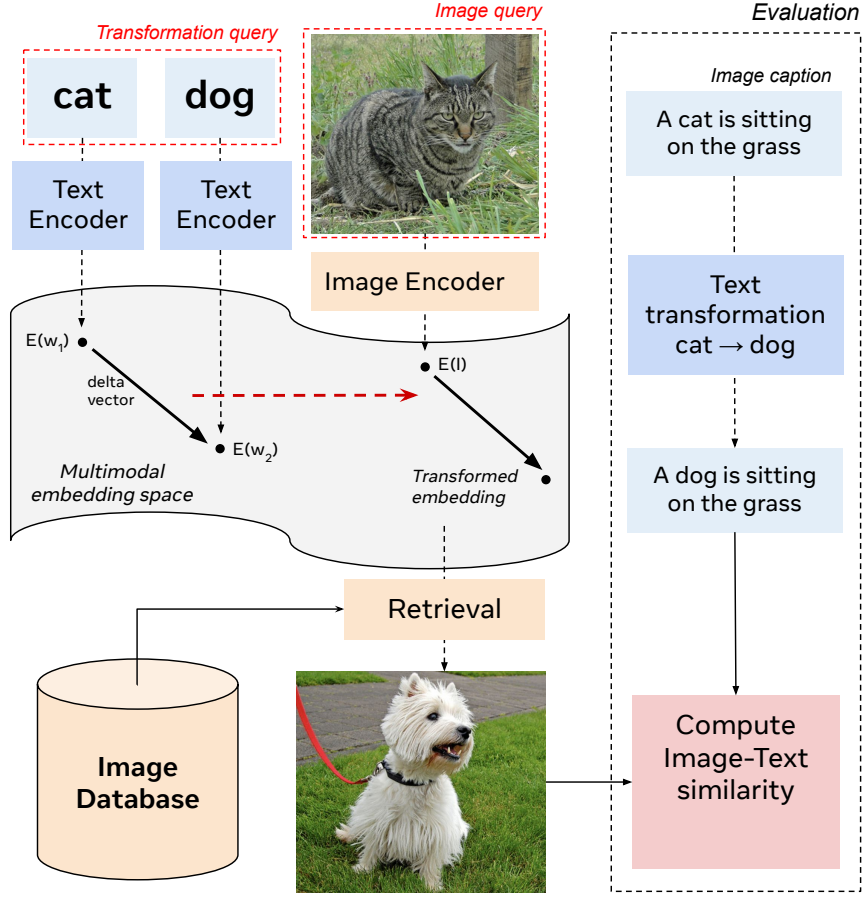


Figure 3.1. – SIMAT image editing with evaluation framework. The transformation is mapped to a *delta vector*, added to the image embedding to produce a *transformed embedding*, for which a corresponding image is retrieved in a database. The evaluation module checks that the text-transformed caption is valid for the image result.

H. Lee 2018), such as the famous analogy: *queen* is to *king* what *woman* is to *man*. Previous work (Jia et al. 2021) has hinted that such geometric properties could also be present in multimodal spaces, without quantitative evidence.

In this chapter, we study the suitability of multimodal embedding spaces like CLIP (Radford et al. 2021a) to perform image retrieval with image-text queries. We also study whether it is beneficial to leverage the geometric properties of sentence embedding spaces to get multimodal embeddings better suited to image retrieval with multimodal queries. In particular, we use LASER (Artetxe and Schwenk 2019) and LaBSE (F. Feng et al. 2020), which have been pre-trained on large corpora of multilingual data.

We aim to define a rigorous evaluation framework for our task. A good dataset to evaluate Multimodal Image Retrieval should contain feasible (image, text trans-

formation) queries: the transformation *man*→*dog* can be applied to an image with “A man is running on the beach”, but not to “A man is speaking on the phone”. We create SIMAT, a corpus based on Visual Genome images and annotations (Krishna et al. 2017) and ensure that this requirement is met. SIMAT contains 6k images and 18k textual transformation queries that aim at either replacing scene elements or changing pairwise relationships between scene elements. Finding metrics for Multimodal Image Retrieval is challenging: first, we need to ensure that the requested transfer is performed (the cat is replaced by a dog). Then, we need to verify that the modification is minimal: the dog should be sitting on grass and ideally, all other visual elements should not be changed. We use OSCAR (X. Li et al. 2020) as an external oracle to assess whether these two conditions are met. OSCAR is a multimodal transformer trained on captioned images with a binary cross-entropy loss to recognize whether a given text corresponds to an image (see Section 2.1 for more details). At the time of writing, it was the best available model for detecting whether a text correctly describes an image.

The remainder of the chapter is organized as follows: we first present in detail how we built the SIMAT database. We then explain our retrieval-based editing framework in the methods section, and finally we conduct a variety of experiments on the SIMAT database.

3.2 The SIMAT database

3.2.1 Existing databases for Multimodal Image Retrieval

Several datasets exist to evaluate Multimodal Image Retrieval on narrow image domains: the CSS dataset (N. Vo et al. 2019b) which is a synthetic dataset with simple colored geometrical objects based on CLEVR (Justin Johnson et al. 2017). The Fashion200k dataset (Han et al. 2017) provides around 200k images of fashion products, each annotated with a compact attribute description. Similarly, the Fashion-IQ dataset (Guo et al. 2019) was built to advance research on interactive fashion image retrieval. The MIT-States dataset (Isola et al. 2015), also commonly used, is a dataset of $\approx 60k$ images, each annotated with an object/noun label and a state/adjective label such as *new car* or *broken window*.

Those datasets are designed to evaluate image retrieval with multimodal queries on narrow domains, which gives more control over what attributes can be changed and ensures that the transformation is always feasible. Another common characteristic is the focus on changing object properties rather than objects relationships. We focus on more realistic images, and study object transformations where an ob-

ject should be replaced by another without changing the high-level subject-object interaction.

3.2.2 Requirements

First, we need a list of images with some transformation queries (e.g. a man sleeping on the beach, with the query *man*→*woman*). We want simple images (so that the query is unambiguous) and relevant transformation queries. Second, we need a database of images that we will use for the retrieval step. And finally, we need a criterion to decide, based on the retrieved image, if the transformation is successful or not. It is the case if only the element designated by the transformation query has changed, while keeping the rest of visual elements as similar as possible.

Previous work (N. Vo et al. 2019a) has tried to solve these requirements with a dataset of $\approx 1,500$ images from Google Image Search queries, dubbed SIC₁₁₂, with each image annotated with an actor-action-environment triplet, such as (*woman, walking, street*), among a set of 112 possible triplets. Transformation queries then consist in changing either the subject, action or environment. This set of images is also used as a database for retrieval, which has two advantages: (i) transformation queries are always possible by design of the dataset, and (ii) the quality of the retrieved image is measured by checking if its annotation triplet is indeed the one expected by the transformation query.

We scale this approach to a larger number of annotation triplets, that take the more general form of (subject, relationship, object). However, we observed that due to the larger triplet vocabulary, images can be accurately described by multiple such triplets, which skews the evaluation metric: an image would often be rejected for not being annotated with the expected triplet while still being visually correct. Therefore, we choose to use a different metric for evaluating the quality of transformed images: we evaluate whether the semantic transformation is successful by querying OSCAR (X. Li et al. 2020). OSCAR computes the probability $\mathbb{P}_O(I, T)$ that a caption T accurately described an image I , based on the concatenation of the text tokens in T and the object tags and features detected by faster R-CNN on image I (we provide the triplet to OSCAR in the form of a caption written in natural language). Note that this OSCAR-based evaluation method does not involve image annotations in the retrieval database and thus could potentially be applied to a much larger database of non-annotated images, which we leave for future work.

3.2.3 Construction

Similarly to (N. Vo et al. 2019a), we create a list of images annotated with (subject, relationship, object) triplets, and perform the retrieval step inside the same list of images to ensure that transformation queries always have a valid solution in the dataset. We start from annotations from the Visual Genome dataset (Krishna et al. 2017). Each image in the dataset contains a list of such triplets with subject and object bounding boxes, which we use to crop square regions of images that minimally contain the subject and object in the image. We then filter this list and compute possible transformations:

Subject/Relation filtering. Only keep triplets for which the subject is a human or animal, and the relation is a non-positional relationship in Visual Genome. The full lists are shown in Figure 3.2.

Object filtering. Only keep objects O for which there exists at least two triplets (S, R, O) and (S', R', O) with $R \neq R'$. This ensures that the selected objects have at least two different types of interaction in images. Then, only keep the 10 most frequent objects for a single (subject, relation) pair. This gives a list of 645 distinct triplets.

Building transformation queries. For each image I with associated triplet (S, R, O) , add in the list of transformation queries $(I, O \rightarrow O')$ if there is a triplet (S, R, O') in the database. Do the same for S and R . This ensures that transformation queries consists of pairs of *objects* that can have the same (subject, relation) pair, and symmetrically for *subjects* and *relations*.

Writing captions for OSCAR. For each of the 645 triplets, we manually wrote a caption in natural language, e.g. $(man, sitting\ on, chair) \rightarrow A\ man\ sitting\ on\ a\ chair$.

We now have a database of images and transformation queries, but we have noticed some noise in the annotation procedure: an image can have a triplet annotation which does not well describe the main action in it, because the cropping procedure included an object that is more important than the extracted triplet. Also, transformation queries sometimes consisted in synonyms. We solve this problem using OSCAR to filter transformation queries: given an image I with query triplet t_1 and target triplet t_2 , we keep the corresponding transformation if $\mathbb{P}_O(I, t_1) > 0.9$ and $\mathbb{P}_O(I, t_2) < 0.1$. This ensures that not modifying the image is not a valid solution to the problem.

The distribution of images being quite skewed (see Figure 3.2), the transformation queries also have a bias towards the more frequent subjects, relations and objects. We alleviate this problem by using re-weighting in the scoring metric (see below).

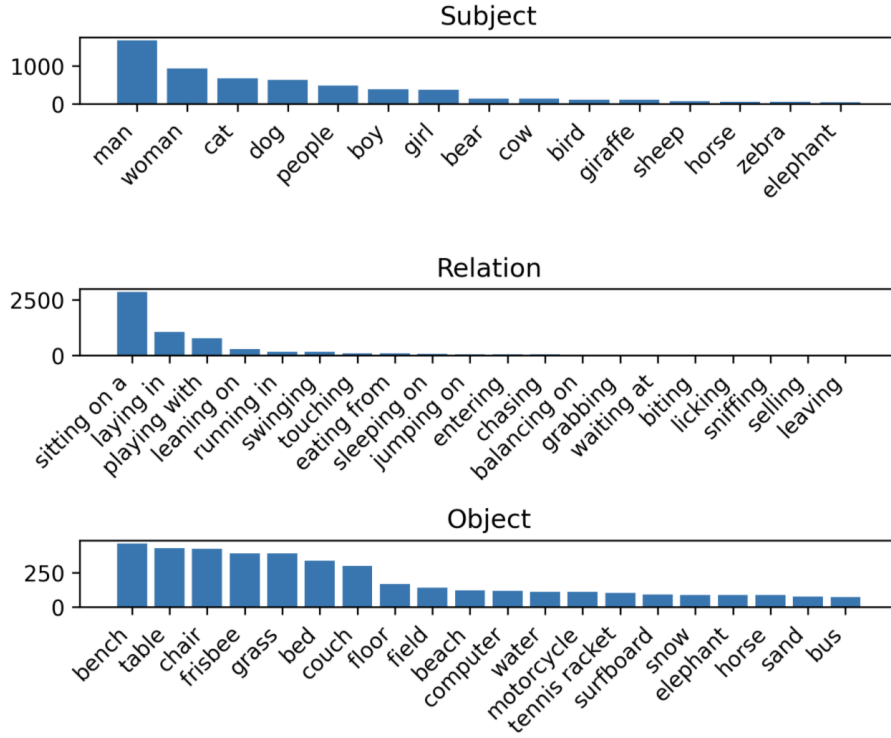


Figure 3.2. – Statistics for SIMAT database. All subjects and relationships are represented, but only 25 objects out of 131 are listed here.

In summary, our SIMAT dataset (for Semantic Image Transformation) consists of:

- 5 989 images, each annotated with a subject-relation-object triplet.
- 17 996 transformation queries on those images, with queries asking to change the subject, the relation, or the object.
- A list of 645 distinct subject-relation-object triplets with corresponding captions, each triplet having at least 2 corresponding images.

To allow hyperparameter selection, we make a 50-50 dev/set split on the list of images, and split the transformation queries accordingly.

3.2.4 Evaluation Metric

Let $(I_i, w_1 \rightarrow w_2, T_i)$ be a sample in our dataset \mathcal{D} where $w_1 \rightarrow w_2$ is the transformation query and T_i is the caption associated to the target triplet of this sample. For this sample, we consider that a retrieved image J_i corresponds to a

successful transformation if OSCAR outputs a probability $\mathbb{P}_O(J_i, T_i) > 0.5$. The final score is simply a weighted accuracy over all dataset samples:

$$S = \sum_{i=1}^{\|\mathcal{D}\|} \mu_i \mathbb{1}_{\mathbb{P}_O(J_i, T_i) > 0.5} \quad (3.1)$$

where the coefficients μ_i are the contributions of each sample to the total score. We adopt an inverse square root re-weighting to down-sample the most frequent transformations.

3.3 Embedding Editing strategies

Starting from semantic transformations in text, we show how text transformations can be transferred to images via multimodal embeddings. We then present our procedure for fine-tuning multimodal embeddings.

3.3.1 Text delta vectors for semantic transformations

Semantic properties in sentences can be modified by word replacement: in the sentence “*A man walking on the beach*”, the semantic property *subject gender* can be changed by replacing the word *man* with the word *woman*. In a latent space, where direct word replacement is not possible, we can apply semantic transformations by doing arithmetic operations. By encoding sentences as the sum of their word embeddings, applying a transformation $w_1 \rightarrow w_2$ on a sentence embedding $E(s)$ amounts to adding the vector $E(w_2) - E(w_1)$, which we call a *delta vector*. In principle, the textual form of the transformed sentence can be found by retrieving the sentence embedding closest to $E(s) + E(w_2) - E(w_1)$ in a database.

However, there is some ambiguity in the process since bag of words representations do not take into account the order of words. That is why we consider more complex non-linear sentence embeddings which have been shown to display similar properties as above (Logeswaran and H. Lee 2018), in addition to better reflecting the meaning of sentences (Artetxe and Schwenk 2019).

We study four sentence embeddings: *CLIP*, obtained by a contrastive loss on a large set of image/text pairs; *FastText*, obtained with a weighted sum of FastText word embeddings (Bojanowski et al. 2017); *LaBSE*, which are trained by matching parallel sentences in different languages with a contrastive loss (F. Feng et al. 2020); and the *LASER* embeddings (Artetxe and Schwenk 2019) which are trained with a multilingual translation task.

3.3.2 From text delta vectors to images

Semantic transformations, seen as *delta vectors* as defined above, can be added to image embeddings in multimodal spaces. As introduced in equation 2.3, we use an image encoder E_{img} and a text encoder E_{txt} that embed both modalities into a shared latent space (see Fig 3.1). The transformed embedding is

$$x = E_{img}(I) + \lambda \cdot (E_{txt}(w_2) - E_{txt}(w_1)) \quad (3.2)$$

The *scaling factor* λ is a hyper-parameter that can be adjusted to increase the strength of the transformation. The natural choice is $\lambda = 1$, but it has been noted that a higher value can help to better enforce the transformation (Jia et al. 2021). The image embeddings are quite sparse due to the relatively small size of the image database, so we found it helpful to enforce the rule that the retrieved image should be different from the input image.

3.3.3 Finetuning multimodal embeddings

We consider multiple choices for the image and text encoders: our default setup is to use the CLIP embeddings for both modalities (63M parameters for the text encoder, 87M for the image encoder), and we experiment with using two ImageNet-pretrained ResNets (ResNet50 and ResNet152, respectively 23M and 63M parameters) as image encoders, and FastText, LASER and LaBSE as text encoders. We can evaluate the vanilla CLIP embeddings without retraining; however, other encoding choices are not directly compatible, and we have to fine-tune the encoders to be able to encode image and text into a shared latent space. We use a very simple fine-tuning scheme on COCO (T.-Y. Lin et al. 2014b) where we train linear adaptation heads after the frozen encoders (Fig 3.3) for 30 epochs with a learning rate of $1e-3$ and a batch size of 4096. Fine-tuning a model takes approx. 3 hours on 8 Tesla V100 GPUs.

When using the ResNet-based encoders, our initial study showed that only training a linear layer is not sufficient to get a reasonable performance on image-text retrieval, because the backbone network is only trained on image classification. Therefore, we freeze only the first three blocks of the ResNet models and add a

simple 4-layer MLP architecture on top of the pooled features. We use an image-text InfoNCE (Sohn 2016a) contrastive loss (which is used for training CLIP):

$$\mathcal{C}(I, T) = \frac{1}{n} \sum_{i=1}^n \left(\frac{\exp(I_i \cdot T_i / \tau)}{\sum_{j=1}^n \exp(I_i \cdot T_j / \tau)} \right)$$

$$\mathcal{L} = \frac{1}{2} \mathcal{C}(I, T) + \frac{1}{2} \mathcal{C}(T, I) \quad (3.3)$$

where I and T are normalized image and text embeddings, τ a temperature parameter which is learnable in CLIP. However, we choose to keep it fixed to study its impact on the transformation score.

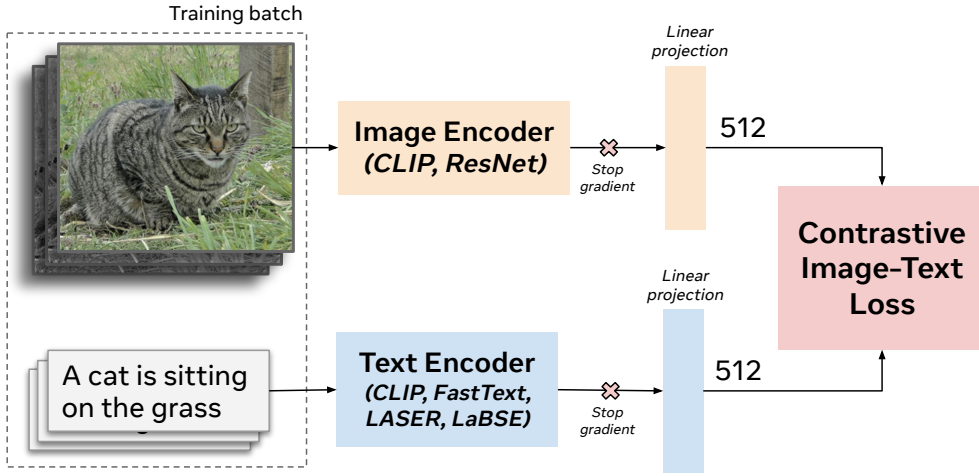


Figure 3.3. – Layer adaptation learning on COCO. The image and text embeddings are projected to a shared multimodal space of dimension 512.

3.4 Experiments

In this section, we analyze the ability of various multimodal embeddings to transfer text transformations to images via *delta vectors*.

3.4.1 Vanilla CLIP embeddings

We first study the performance of the vanilla CLIP embeddings for transferring text transformations to images with delta vectors. To put our results in perspective, we also evaluate the following baselines:

Text to Image: we directly provide the target captions to the CLIP text encoder and retrieve the image closest to that embedding. This is the standard image





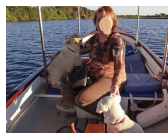





Image Query					
Transformation Query	Woman → Man	Leaning on → Jumping over	Toilet → Suitcase	Kite → Rail	Boat → Bed
Target Caption	A man balancing on a surfboard.	A horse jumping over a fence.	A cat sitting on a suitcase .	A man leaning on a rail .	A woman sitting in a bed .
Retrieved Image					
Success (OSCAR)	YES	YES	YES	NO	NO

Figure 3.4. – Transformation examples from the CLIP model fine-tuned on COCO with temperature $\tau = 0.1$. Columns 1-3 show examples of successful subject, relation and object transformations. Column 4 shows an example of an unsuccessful object transformation: the retrieved image contains a bench instead of a rail, but we can note some visual similarity with a rail. Row 5 shows a frequent mode of failure: the object is the correct, but the relation has been modified. We assume that our algorithm prioritized keeping the dog in the image.

retrieval task, which is easier because the target subject-relation-object features are given as input. , it can be considered as an upper bound of our SIMAT score.

Image to Text to Image: we first find among the SIMAT captions, the text embedding that is closest to the query image. We then add the text delta vector corresponding to the transformation query and finally retrieve the closest image in the SIMAT database. This means that we do not transform the input image directly, but we transform a textual representation of the input image.

Results are shown in Table 3.1. The delta vector method works for 15.9% of the transformation queries. A higher value of λ gives much better results (35.4%) which are nonetheless below the Image to Text to Image baseline (39%), and very far from the Text to Image upper bound (65.9%). It means that with our benchmark, transforming images works better by using text representations of images rather than the image embeddings themselves. However, in a real-world scenario, we don't want to get explicit context of images by converting them to text (which requires a form of image captioning); we want to use the image embeddings as implicit context.

Method	SIMAT score	
	$n = 1$	$n = 5$
Delta Vectors ($\lambda = 1$)	15.9	39.2
Delta Vectors ($\lambda = 3$)	35.4	67.6
Image to Text to Image	39.7	71.0
Text to Image	65.9	95.6

Table 3.1. – SIMAT score for delta vectors in the original CLIP multimodal space. The default score considers the nearest neighbor in the retrieval step ($n = 1$). We also report the SIMAT score for the best image using $n = 5$ nearest neighbors.

3.4.2 Fine-tuning CLIP on COCO

In this section, we consider CLIP as image and text encoder, but we additionally train adaptation layers on COCO with different values for the temperature parameter τ . Figure 3.5 shows the SIMAT score as a function of the scaling factor λ , on the SIMAT dev set. The same curve for the vanilla CLIP embeddings is shown in black. We can see that all curves have an optimal value for λ , which depends on τ . This optimal value $\lambda^*(\tau)$ decreases as τ increases from 0.01 to 1, and the global optimum is reached for $\tau = 0.1$ and $\lambda = 1$. For these values, the SIMAT score is 48.2 which is a 33% absolute improvement over the zero-shot score.

We therefore conclude that the temperature parameter τ has a great importance for transferring text delta vectors to images, and that the fine-tuned embeddings work best with delta vectors for $\tau = 0.1$ and $\lambda = 1$.

Here, we make the important observation that the empirical optimal value for λ is exactly the theoretical value of 1 that should be used to transform bag of word embeddings. Given that $\lambda = 1$ is suboptimal for vanilla CLIP embeddings, we make the hypothesis that multimodal embeddings that are optimal for $\lambda \neq 1$ can be projected to embeddings better suited for delta vectors (hence having better geometric regularities) that maximize transformation accuracy for $\lambda = 1$.

Transformation examples on SIMAT obtained with this model are presented in Figure 3.4.

Note that the best image retrieval and text retrieval evaluations on COCO are obtained for $\tau = 0.01$, which hints towards the fact that smaller temperatures are better for image-text retrieval and higher temperatures ($\tau \sim 0.1$) are more compatible with the delta vector framework. In the rest of this chapter, we use a fixed temperature of $\tau = 0.1$.

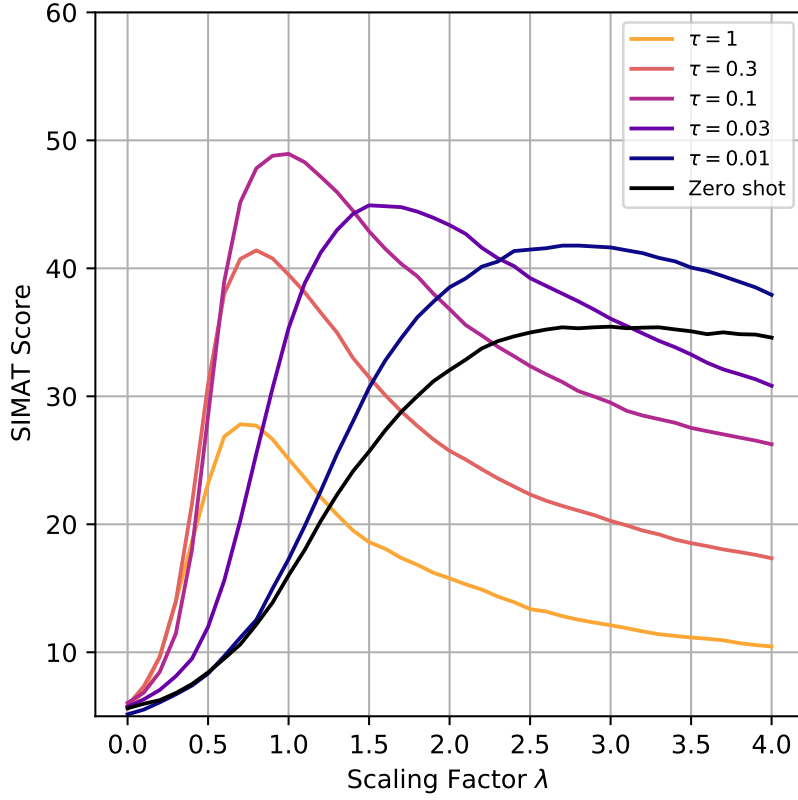


Figure 3.5. – SIMAT score as a function of the scaling factor λ (on development set). The overall best score is obtained for $\tau = 0.1$ and a scaling factor of exactly 1.

3.4.3 Using pretrained text encoders

We show in Figure 3.6, that a value of $\tau = 0.1$ which is optimal for CLIP, is also near-optimal for all other considered text embeddings, FastText, LASER and LaBSE. It seems to be a value that works well for delta vectors. In Table 3.2, we analyze our different choices for the image and text encoders. The *Retrieval upper bound* metric corresponds to the Text to Image baseline of Section 3.4.1. The *Text delta vector* metric is an evaluation of how well the text-defined delta vector can accurately transform the caption of the input image (and not the image itself). We also compute the standard image/text retrieval metrics (Image R@1 and Text R@1) on the COCO test set.

We can see that the key contributing factor in the different SIMAT scores is which sentence encoder has been used. If we fix the sentence encoder, the image encoder has an important influence on the image-text retrieval metrics but

Image Encoder	Sentence Encoder	MS Coco		Text	SIMAT score	Retrieval upper bound
		Text R@1	Image R@1	delta vectors		
RN50	CLIP	25.4	22.3	88.0	44.5	76.2
RN152		27.6	23.5	87.2	46.0	77.7
CLIP		45.2	34.8	82.4	48.2	75.4
RN50	FastText	17.6	15.2	95.3	44.6	65.6
RN152		19.1	16.3	95.5	46.7	68.0
CLIP		28.2	21.9	94.4	47.5	70.6
RN50	LaBSE	18.8	16.8	91.0	38.8	66.9
RN152		20.4	17.9	90.7	39.9	69.0
CLIP		31.4	24.9	92.9	41.9	69.9
RN50	LASER	17.0	15.4	92.1	37.0	67.0
RN152		19.0	16.9	92.6	36.0	67.6
CLIP		29.6	22.8	92.8	37.7	67.6

Table 3.2. – Comparison of different image and sentence encoders for the evaluation of delta vectors ($\tau = 0.1$).

very little impact on the SIMAT score. Therefore, we conclude that improving multimodal embeddings at the task of image/text retrieval will not necessarily improve their geometric properties (in the context of delta vectors).

Also, quite unexpectedly, the SIMAT score does not seem correlated to the Text delta vector score, which measures how well delta vectors can transform text embeddings: the fine-tuned CLIP text embeddings have a text transformation accuracy of 82.4% whereas the fine-tuned FastText embeddings reach 94.4%. Yet they have very similar SIMAT scores (48.2% vs 47.5%). It seems to show that within our constraints, a slightly lower performance on text delta vector (which indicates an embedding space with less geometric structure on the text side) is not the current limitation.

3.4.4 Sentence-based delta vectors

In our default method for using text-based delta vectors, we used single words as input to the text encoder. This is particularly well suited for the FastText embeddings which are based on word embeddings, but not so much for the LASER and LaBSE sentence encoders which are built to encode sentences and not single words. This could explain the performance gap between FastText and LASER/LaBSE. To test this hypothesis, we changed our definition of delta vectors

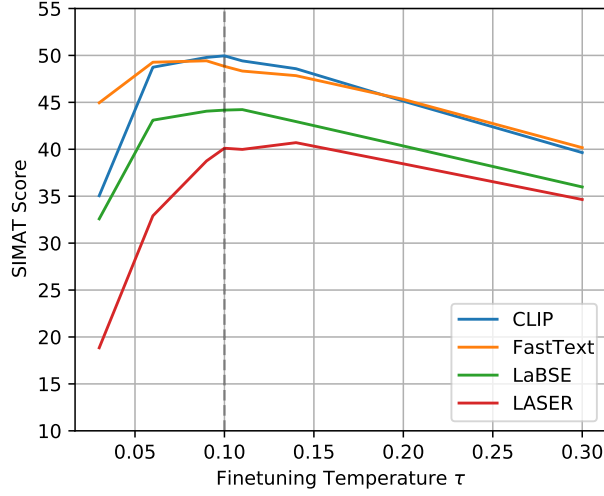


Figure 3.6. – SIMAT score ($\lambda=1$) as a function of training temperature, for several text encoders. For all, the maximum SIMAT score is always obtained for $\tau \approx 0.1$.

so that it is computed by encoding sentences rather than single words. We define the *sentence average delta vector* of transformation $w_1 \rightarrow w_2$ as the average of delta vectors $E(s_2) - E(s_1)$ where s_1 and s_2 go over all pairs of SIMAT captions such that s_2 is the result of the text transformation $w_1 \rightarrow w_2$ applied to s_1 .

We show the results in Table 3.3. With this new method, the performance gap between the different text encoders is much smaller, the SIMAT score being higher for LASER and LaBSE, and smaller for FastText. We observed that we can use a higher scaling factor to boost the SIMAT score, up to $\lambda = 1.5$ for CLIP. We suspect this is due to the fact that the second method produces more reliable delta vectors with a smaller norm.

Note that the role of this experiment is to shed light on the reasons behind the performance spread with respect to the text encoders. The captions of SIMAT should be reserved for evaluation only and not used within the algorithm. A better algorithm may use the COCO captions to create better sentence-based delta vectors, but we leave this for future work.

3.4.5 Transformation score by target

In Figure 3.7, we use the CLIP model fine-tuned with $\tau = 0.1$ and compute the SIMAT score by grouping transformations by their target values: for each word w , we compute the weighted accuracy of transformation success for all queries $w_1 \rightarrow w_2$ such that $w_2 = w$. The relative weight of each target value in the final

Sentence Encoder	Single word	Sentence Average		
		$\lambda = 1$	$\lambda = 1.2$	$\lambda = 1.5$
CLIP	48.2	46.7	51.5	53.5
FastText	47.5	44.6	46.5	45.8
LASER	37.7	43.8	45.0	44.2
LaBSE	41.9	44.6	46.5	45.5

Table 3.3. – Comparison of two methods to calculate delta vectors: *Single word* and *Sentence average*. With the latter, all the encoders have very similar SIMAT scores.

SIMAT score is shown on the x-axis, and the y-axis represents the SIMAT score. We can see that overall, transforming object relations is harder than transforming the objects themselves, which is probably because relationships are less easily identifiable in images. Also, if we compare the SIMAT scores between objects, we can see that the best SIMAT scores are obtained for objects that are easy to recognize (sink, toilet, suitcase...) while the worst scores correspond to objects without a well-defined shape that are harder to recognize (feeder, counter, wall, bus stop).

3.5 Conclusion

In this chapter, we introduced SIMAT, a novel dataset to study the task of text-driven image transformation. It is much larger and has a wider variety of transformations than existing approaches like SIC₁₁₂. Due to this larger size, we argue that evaluation cannot be performed solely by using the caption of the retrieved image, and we propose to use OSCAR to assess whether an input image has been successfully transformed.

We use SIMAT to study the geometric properties of multimodal embedding spaces trained with an image-text alignment objective. We use a simple linear approach (*delta vectors*) for transferring text-defined transformations to images in multimodal spaces, which should work well for well-structured spaces. This provides a novel way to study multimodal embedding spaces compared to standard image/text retrieval metrics in the literature.

After having evaluated vanilla CLIP multimodal embeddings, we have studied embeddings obtained by training with an image/text alignment on COCO, that use pretrained text encoders (FastText, LASER, LaBSE) and pretrained image encoders (CLIP, Resnet50, Resnet152). We emphasize below our findings:

LASER and LabSE is actually harmful, but we show that this is almost entirely due to the fact that we only have access to single words to compute the text delta vector.

While this work provided interesting insights on the regularities of multimodal embedding spaces, it does not provide real image editing and is inherently limited by the database’s size. In the remainder of the thesis, we study how to directly modify the input image as requested by the transformation query. Unlike database retrieval, this requires (i) to identify parts of the image that need editing, (ii) to synthesize novel image parts, and (iii) to blend them seamlessly with the unmodified other image parts. In the following chapters 4 and 5, we propose two different approaches to these challenges.

IMAGE EDITING WITH INSTANCE-BASED OPTIMIZATION

4.1 Introduction

In the previous chapter, we have studied a simplified editing problem based on database retrieval instead of really editing the input image. Real editing requires strong image synthesis skills, and should ideally allow for a wide variety of edit operations that can be described in natural language. While Generative Adversarial Networks (GANs) hold the state-of-the-art in image synthesis on structured domains (such as human faces), scaling them for text-to image generation and editing is still very challenging.

In this chapter, we propose a more flexible approach to Semantic Image Editing, dubbed FLEXIT. Given an input image and a user defined text query of the form (*SaT*) (like *cat* \rightarrow *dog*), we define a multimodal embedding "target point" that represents the edited image, similarly to what we did in Chapter 3. The target point is a linear combination of the inputs CLIP embeddings (Radford et al. 2021b), which contains one image embedding and two text embeddings for the source and target texts. However, instead of searching for a corresponding image in a database, we perform a per-image optimization procedure that gradually makes changes in the input image to get closer to the multimodal target embedding. The image is optimized in the latent space of the VQGAN image auto-encoder, presented in Chapter 2. Compared to pixel space optimization, it allows us to avoid adversarial optimization and to get edited images closer to the manifold of real images, improving image realism. Compared to GAN latent space optimization, it allows us to process a much wider variety of images than that used for training the GAN generative model. A few editing examples are shown in figure 4.1. Using the CLIP networks, pretrained on web-scale data, lets us process a wide variety of textual transformation queries, thanks to the well-organized CLIP multimodal latent space. We also propose a variety of regularization strategies to ensure image quality and relevance to the transformation query. FLEXIT requires only fixed pretrained components, and can thus be used off-the-shelf without



Figure 4.1. – FLEXIT transformation examples. From top to bottom: input image, transformed image, and text query.

requiring any training, which is essential in the context of image editing where training data is very scarce (see Chapter 2 for details).

The lack of examples for the semantic image editing task also means that there is no satisfying dataset for evaluation. Most previous works in the field have focused on qualitative examples, and the available relevant quantitative evaluations only covered very few textual transformation queries. To remedy this problem, we propose a quantitative evaluation protocol for the task of semantic image editing. It is based on the three main editing criteria presented in Chapter 2, that we recall here: (i) the transformed image should correctly correspond to the text query, (ii) the output image should look natural, and (iii) visual elements irrelevant to the text query should remain unchanged. We thoroughly evaluate our model on ImageNet, and demonstrate quantitatively and qualitatively the superiority of our method against baselines.

The remainder of the chapter is organized as follows: after going over related work, we present the FLEXIT algorithm for Semantic Image Editing. We then conduct experiments with our ImageNet-based evaluation protocol, and ablate design choices in our methods, especially focusing on how to set values of hyperparameters for optimal editing. We conclude by underlying the benefits of the method as well as its limitations.

4.2 Related Work on Image Editing in latent spaces

In this section, we go over different methods for editing images in GANs latent spaces, and we give the motivation for using the VQ-GAN latent space, coming from an image auto-encoder instead of a GAN training.

Image editing with GANs Modern Generative Adversarial Networks like StyleGAN (Tero Karras et al. 2019; T. Karras et al. 2020; Tero Karras et al. 2021), have an impressively disentangled latent space, where performing copy-pastes between two latent vectors transfers the corresponding styles in the image space. Consequently, significant research efforts have been put into using pretrained GANs for semantic image edition, as mentioned in paragraph 2.3.

More precisely through specific latent-space manipulation, high-level attributes such as age or gender can be identified and edited realistically (Shen et al. 2020b; Abdal et al. 2021; Zhuang et al. 2021; Härkönen et al. 2020). By using an auxiliary classifier, a simple approach consists in finding linear boundaries in the latent space separating binary attributes (Shen et al. 2020b; Zhuang et al. 2021; Goetschalckx et al. 2019), which allows editing attributes by “walking” in the orthogonal latent direction. StyleFlow (Abdal et al. 2021) proposes a non-linear approach by learning the latent transformations using normalizing flows. Other methods (Härkönen et al. 2020; Voynov and Babenko 2020) operate without a pre-trained classifier and find the transformations in an unsupervised manner, requiring a manual labelling process to interpret and annotate the “discovered” transformations. Finally, StyleCLIP (Patashnik et al. 2021) guides latent editing with a multimodal objective in a CLIP space. These approaches, however, present several caveats. First, contrary to generated latents, inferred latent codes representing real images have been shown to react poorly to latent editing operations (Grechka et al. 2021a). Moreover, edit operations are also limited to the semantics identified in the latent space, which are specific to the single domain the GAN was trained on, such as age or apparent gender in the case of faces.

Another line of research for image editing is *image-to-image translation*, which consists in training a GAN to directly modify the images. These methods learn a transformation between two domains, using paired data (Isola et al. 2017; T.-C. Wang et al. 2018; Taesung Park et al. 2019) or unpaired data (J.-Y. Zhu et al. 2017; Y. Choi et al. 2020). In ManiGAN, B. Li et al. (2020a) train an image-to-image translation GAN to increase semantic consistency with text caption in images. However, these models only learn a single transformation, or combinations thereof (Y. Wang et al. 2020), specific to the training data, limiting the scope of their applicability. Rather such restricted sets of possible edit dimensions, we target more general transformations described by free-text.

Image latent space. While GANs are highly effective as generative models, inference of the latent variable given an image is in principle intractable. Even though joint learning of an inference network has been proposed, see *e.g.* (J. Donahue et al. 2017; Dumoulin et al. 2017), the mode-seeking training dynamics of

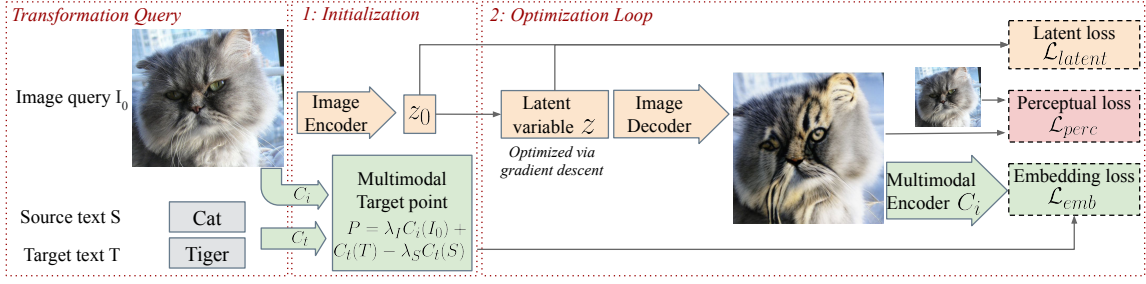


Figure 4.2. – FLEXIT optimization framework: components involving the multimodal latent space colored in green; those involving the image latent space in yellow; those involving the LPIPS distance in pink. Given a transformation query (I_0, S, T) , we first compute a target point P in the multimodal embedding space, and we encode I_0 in the image latent space to get z_0 . Then, for a fixed number of steps, we update the latent variable z (initialized with z_0) to get closer to the target point P . We add two regularization terms: the LPIPS perceptual distance between the input image and the output image, and a latent distance between z and z_0 . All networks are frozen, only z is updated.

GANs are not suited for good reconstruction performance beyond the training distribution (or even within it, if modes are dropped).

Variational autoencoders (Kingma and Welling 2014), on the other hand, offer an inference network by construction, and their likelihood-based training objective ensures accurate reconstructions. Vector-quantized variational autoencoders (VQ-VAE) (A. van den Oord et al. 2017; Razavi et al. 2019), which discretize the latent space, have been found to offer both good reconstructions and compelling samples. In this paper, we use the VQ-GAN variant (Esser et al. 2021b) presented in Chapter 2 which includes an adversarial loss term to train the autoencoder.

4.3 FlexIT algorithm

An overview of our image transformation approach is depicted in Figure 4.2. It relies on three pre-trained components. First, we edit the input image in a latent space, with the requirement that a wide range of images can be encoded and decoded back to an RGB image with minimal distortion. We chose the VQGAN autoencoder (Esser et al. 2021b) for that purpose. Second, we embed the text query and input image in a multimodal embedding space, to define the optimization target for the modified image. We use the CLIP (Radford et al. 2021b) multimodal embedding spaces. Finally, to ensure that the modified image remains similar to the input, we control its distance to the input image with the LPIPS perceptual

distance (R. Zhang et al. 2018) computed with a VGG (Simonyan and Zisserman 2015) backbone.

Optimization scheme. The core idea of the FLEXIT method is to edit the input image in a latent space, guided by a high-level semantic objective defined in the multimodal embedding space. Let E be the image encoder, D the image decoder and (C_t, C_i) the multimodal encoders for text and image respectively. Given an input image I_0 and a textual transformation $S \rightarrow T$, we first initialize FLEXIT by computing the initial latent image representation as $z_0 = E(I_0)$ and the target multimodal point P as

$$P = C_t(T) + \lambda_I C_i(I_0) - \lambda_S C_t(S). \quad (4.1)$$

We choose to use a multimodal embedding space since it allows text and image modalities to be combined in a meaningful way: semantic transformations defined by textual embeddings can be applied to images with linear operations (Jia et al. 2021). In this context, our target point P can be seen as an image embedding that has been semantically modified with textual embeddings, by removing the source class information $(-\lambda_S E_t(S))$ and adding the target class information $(+E_t(T))$. Equation 4.1 is indeed similar to equation 2.3 with additional hyperparameters: since we don't know what is the optimal linear combination of image and text embeddings, we consider λ_I and λ_S as parameters which will be validated on our development set.

To find an output image which, when encoded in the multimodal embedding space, gets as close as possible to the target point, we optimize the embedding loss:

$$\mathcal{L}_{emb}(z) = \|C_i(D(z)) - P\|_2^2, \quad (4.2)$$

which is similar to the objective introduced in equation 2.4 from Chapter 2. We add two regularization terms to the embedding loss, to encourage that only the content related to the transformation query is changed. Without regularization, the optimization scheme can alter any part of the image if this helps in getting closer to the multimodal target point, which we have found to yield unnatural artifacts. The distance to the input image I_0 is controlled with a LPIPS distance:

$$\mathcal{L}_{perc}(z) = d_{LPIPS}(D(z), I_0). \quad (4.3)$$

To enforce staying in parts of the latent space that are well decoded by our image decoder, we use a regularization term with respect to the initial latent code

z_0 . We use an ℓ_2 norm at each spatial position i of the latent code, and sum these norms across spatial positions to obtain the loss:

$$\mathcal{L}_{latent}(z) = \sum_i \|z^i - z_0^i\|_2. \quad (4.4)$$

This $\ell_{2,1}$ loss encourages sparse z^i changes, *i.e.* limiting changes in spatial locations, which is aligned with our objective to transform a localized part of the input image.

Finally, note that λ_I in Equation (4.1) also acts as a regularization parameter, by encouraging the input and output image to be close in the multi-modal embedding space.

The total loss we optimize can be written as:

$$\mathcal{L}_{total}(z) = \mathcal{L}_{emb}(z) + \lambda_p \mathcal{L}_{perc}(z) + \lambda_z \mathcal{L}_{latent}(z). \quad (4.5)$$

After initialization, the latent image variable z is updated via gradient descent with a fixed learning rate μ for a fixed number of steps N , while keeping all network weights frozen. Following the implementation of the Fast Gradient Method (Y. Dong et al. 2018), we normalize the gradient before the update.

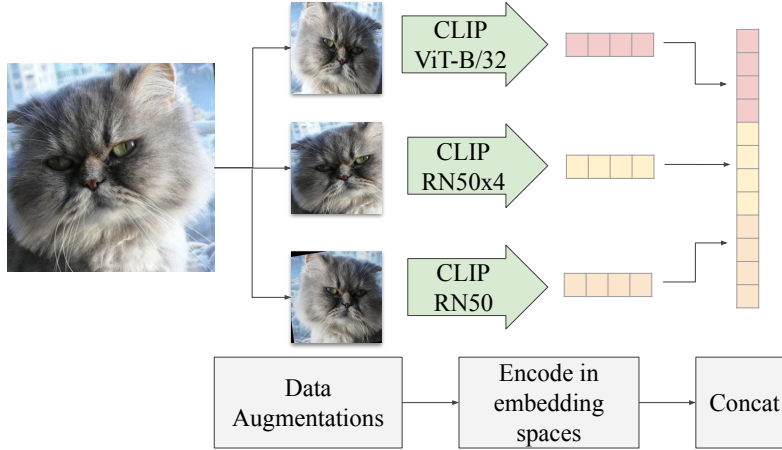


Figure 4.3. – Architecture of our robust CLIP-based image encoder, which combines three different encoders by concatenation.

Image optimization space. The distance to the multi-modal target point is a differentiable loss that can be optimized via gradient descent. A straightforward approach consists in performing gradient descent directly in the pixel-space. However, this type of image representation lacks a prior on low-level image statistics. By optimizing over a latent variable instead, the image is obtained as the

output of a neural-network based decoder. Choosing an autoencoder, like that of VQGAN, lets us (i) make use of the decoder’s low-level priors, which guides the optimization problem towards images that exhibit at least low-level consistency; and (ii) encode and decode images in its latent space with little distortion. The spatial dimensions in the VQGAN latent space allows editing specific parts of the image independently, contrary to GANs which typically rely on more global latent variables. Although GANs generate realistic images with stronger priors, it is problematic to optimize their latent space for two reasons: first, GANs work well on narrow distributions (such as human faces), but do not work as well when trained on a much wider distribution; second, even with a GAN trained on a wide distribution such as that of ImageNet, it is hard to faithfully reconstruct an image using its latent space.

We report on experiments with optimization over raw pixels and GAN latent spaces in Section 4.4.3.

Implementation details. In FLEXIT, we run the optimization loop for $N = 160$ steps, which we found enough to transform most images. We use a resolution of 288 for encoding images with VQGAN, which compresses the images in a latent space with dimensions (256, 18, 18).

We take advantage of various pre-trained CLIP models, and combine their embeddings with concatenation, as shown in Figure 4.3. By default, we use three image embedding networks with different ResNet and ViT architectures (RN50, RN50x4, ViT-B/32), which implement complementary inductive biases. To encode an image with a single CLIP network, we average the embeddings of multiple augmentations of the input image (8 by default). We use a random horizontal flipping and a random rotation between -10 and 10 degrees, followed by cropping the image (keeping at least 80% of the input image) with aspect ratio between 0.9 and 1.1. We have empirically observed that using multiple augmentations per network stabilizes optimization in the early stages.

For the regularization coefficients, we use $\lambda_z = 0.05$, $\lambda_p = 0.15$, $\lambda_S = 0.4$, $\lambda_I = 0.2$ as our default values. These coefficients are set using our ImageNet-based development set, and are fixed for all experiments.

These implementation choices are analyzed in Sec. 4.4.4.

4.4 Experiments

Below, we first describe our evaluation protocol in detail. We then present qualitative and quantitative results, and an in-depth analysis of various components of our approach.

4.4.1 Evaluation Protocol

Evaluation dataset. We did not find a satisfying evaluation framework to study the problem of semantic image translation: existing dataset and metrics focus on narrow image domains, or random text transformation queries (B. Li et al. 2020a; Patashnik et al. 2021). To overcome this, we have decided to build upon the ImageNet dataset (J. Deng et al. 2009b) for its diversity and its high number of classes: by defining which class labels can be changed into one another (like *cat* a *tiger*), we can build a set of sensible object-centric transformation queries. We have selected a subset of the 273 ImageNet labels that we manually split into 47 clusters according to their semantic similarity. For instance, there is a cluster containing all kinds of vegetables. Details on the subset selection and grouping are presented in the appendix. We only consider transformations $S \rightarrow T$ where S and T are in the same cluster, in order to avoid nonsensical transformations between unrelated objects, *e.g.* laptop a butterfly.

For each target label T we construct eight transformation queries by randomly sampling eight other classes $\{S_i\}$ within the same cluster, and sample a random image from each S_i from the ImageNet validation set. This gives a total of 2,184 transformation queries that we split into a development set and a test set of equal size. We use the development set to tune various hyper-parameters of our approach, and report evaluation metrics on the test set.

Measuring transformation success. We evaluate the success of the transformation by means of the **Accuracy** of an image classifier, which is possible since we use ImageNet class labels as the transformation targets. We use a DeiT (Touvron et al. 2021) classifier, which has an ImageNet validation accuracy of 85.2%. We judge a transformation successful if, for the transformed image, class T has the highest probability among the 273 selected classes. This metric is similar to the OSCAR metric, introduced in Chapter 3, since it is a binary measure of whether the transformed image actually corresponds to the target text.

Measuring image realism. To assess naturalness of transformed images, we use a variant of the Fréchet Inception distance (FID) (Heusel et al. 2017a), which measures the distance between the distributions of the real images and generated images in the feature space of an InceptionV3 classifier C. Szegedy et al. 2016. Since the images we transform are extracted from the ImageNet validation set, we use the ImageNet training set as our reference distribution. To avoid numerical instability related to estimating the feature distribution with a small number of samples, we use the “Simplified FID” (**SFID**) (C.-I. Kim et al. 2020) which does not take into account the off-diagonal terms in the feature covariance matrix. In

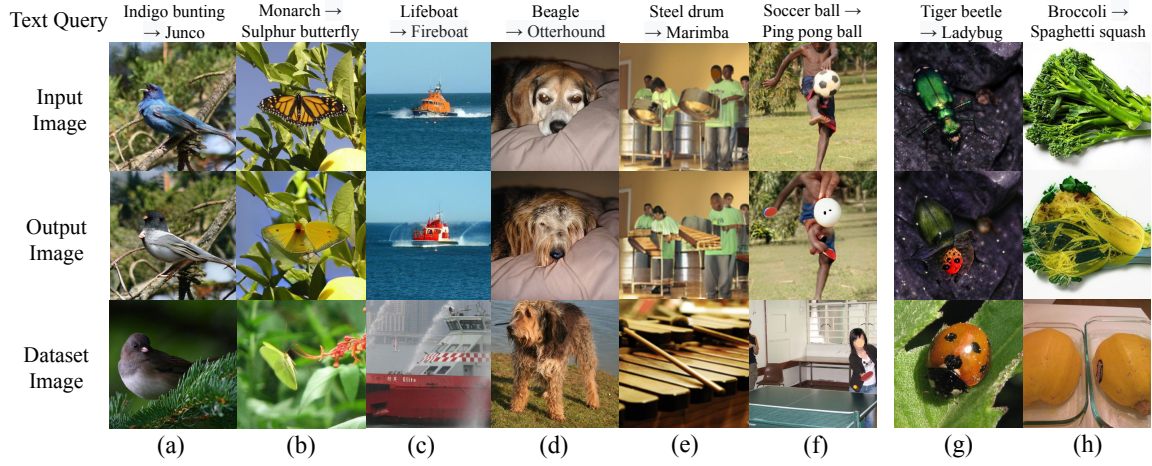


Figure 4.4. – Transformation examples with FLEXIT on ImageNet images. From top to bottom: input and output image, as well as dataset image from the target class. Columns (a)-(e) show examples of successful transformations. Column (f) shows an interesting behavior where another object has been added in the image to add more context (a table tennis racket in the hand of the person). The last two columns show the most frequent modes of failure: only part of the input object is transformed (g), or parts of the input object that should be changed are not changed: in column (h), the transformed images still has a broccoli shape with green parts instead of an orange and round spaghetti squash.

addition to the SFID, we use a class-conditional SFID score (**CSFID**) which is an average of the SFID scores computed for each target class separately.¹ Because we compute these scores with a low number of examples for many classes, the CSFID score has a high bias, low variance profile on our dataset (Chong and Forsyth 2020), and we have found it to be reliable and stable. The CSFID metric is a measure of both image quality and transformation accuracy, as it measures the feature distribution distance between the transformed images and the reference images from the target class in the training set.

Measuring image distortion. Editing should not change parts of the image that are irrelevant to the transformation defined in the text, *e.g.* the background. We use the **LPIPS** perceptual distance (R. Zhang et al. 2018) to measure deviation from the input image. It is a weighted ℓ_2 distance of deep image features, and has been demonstrated to correlate well with human perceptual similarity. During training, we used the LPIPS distance based on VGG features, to reduce bias in the LPIPS evaluation which is based on AlexNet features A. Krizhevsky et al. 2012.

1. Referred to as within-class FID in (Benny et al. 2021).

For both training and evaluation, LPIPS scores are computed at resolution 256. The LPIPS distance cannot differentiate between edits that are relevant to the text query, and those which are not; and we don't know the minimal LPIPS distance between an image and its closest successful transformation. Still, we argue that it should be as low as possible.

4.4.2 Results

Qualitative results of FLEXIT transformations on ImageNet images are presented in Figure 4.4, including successful transformations as well as several failure cases.

Figure 4.5 shows intermediate transformation results with FLEXIT for 0, 8, 16, 32 and 160 optimization steps. The result after zero optimization steps shows the effect of autoencoding the input image, without changing the latent representation.

We also show examples of color transformations for images from the Stanford Cars dataset (Krause et al. 2013) in Figure 4.6.



Figure 4.6. – Example transformations on the *Cars* dataset: input images (first row), FLEXIT results (second row), StyleCLIP results based on a StyleGAN2 backbone pre-trained on LSUN Cars dataset (last row). Although GAN-based images have better details like the wheels, they are farther away from the input images.

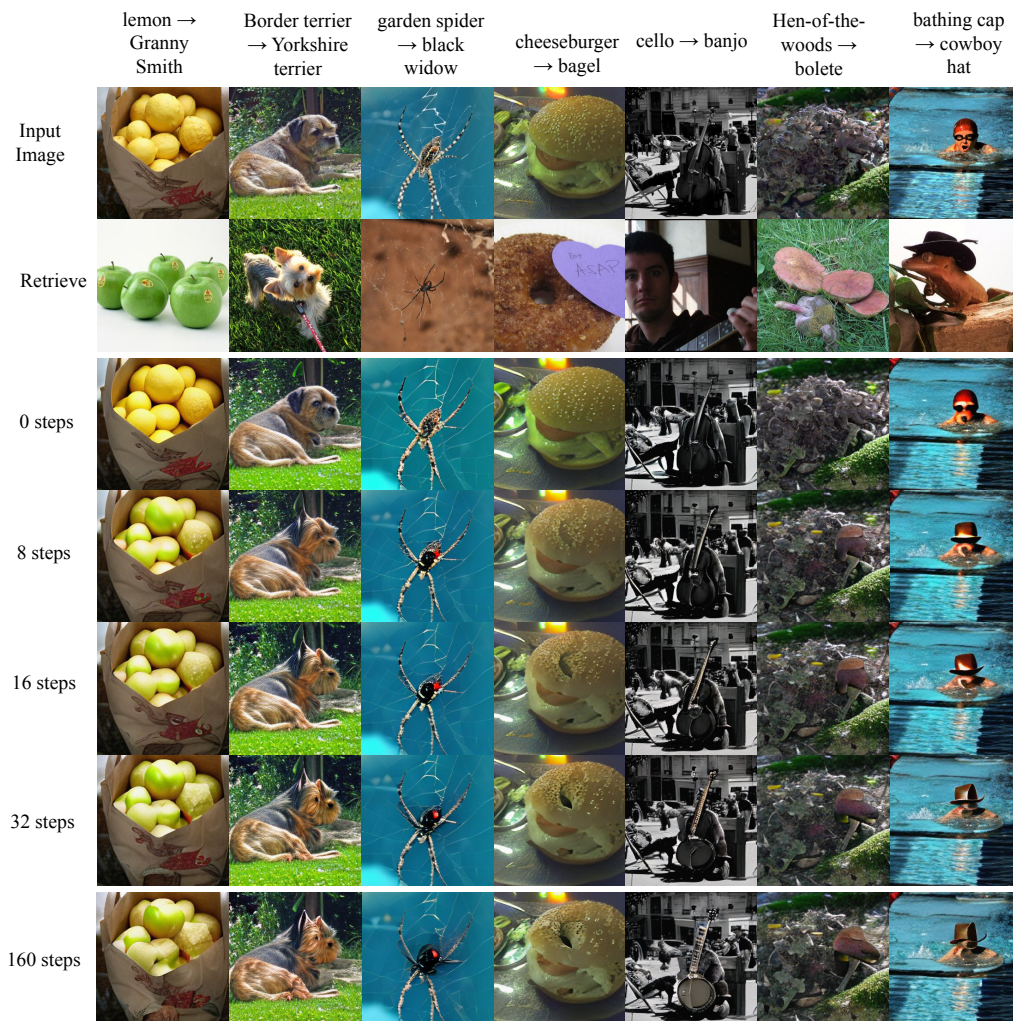


Figure 4.5. – Intermediate transformation results obtained with FLEXIT . Note that most edits only require 32 steps to be completed; some edits benefit from longer optimization schemes, such as the spider and the banjo.

	LPIPS ↓	Acc.%↑	CSFID ↓	SFID ↓
COPY	0.0	0.45	106.0	0.2
ENCODE	17.5	1.6	107.5	3.0
RETRIEVE	72.4	90.6	27.2	0.2
ManiGAN (B. Li et al. 2020a)	21.7	2.0	123.8	17.0
StyleCLIP (Patashnik et al. 2021)	33.4	8.0	146.6	35.8
FLEXIT (Ours)	24.7	51.3	57.9	6.8

Table 4.1. – Evaluation of FLEXIT and baselines on ImageNet images.

Semantic image translation is inherently a trade-off between having the most relevant and natural output image (as measured by Accuracy, CSFID and SFID), while staying as close as possible to the input image (as measured by LPIPS). We consider two extreme configurations as baselines, which only optimize one of these two criteria: (i) The COPY baseline, which simply copies the input image without any modification, and (ii) the RETRIEVE baseline that outputs a random validation image labelled with the target class T . We add the ENCODE baseline that simply passes the input image through the VQGAN autoencoder.

We compare FLEXIT against StyleCLIP (Patashnik et al. 2021), a similar text-driven image transformation algorithm from the literature. We consider the version most similar to our method that embeds images with an ImageNet-trained StyleGAN2,² and iteratively updates the StyleGAN2 latent representation to maximize the similarity with a given text in the CLIP latent space. We have also trained ManiGAN (B. Li et al. 2020a) on ImageNet with the implementation from the authors.

Results are reported in Table 4.1. As expected, the copy baseline is ideal on LPIPS and SFID, but fails to adapt to the transformation target T , and thus fails on Accuracy and CSFID. For the same reason, the auto-encoding baseline also fails on Accuracy and CSFID, but demonstrates the non-trivial impact of using the VQGAN latent space on LPIPS and SFID. The RETRIEVE baseline provides ideal metrics for Accuracy, CSFID and SFID, as it returns natural images of the target class. It fails on LPIPS, however, since the output image is unrelated to the input.

Our FLEXIT approach combines a low LPIPS (24.7 v.s. 17.5 for ENCODE) with an accuracy of 51.3% and a CSFID of 57.9, which is closer to the CSFID of RETRIEVE (27.2) than that of ENCODE (107.5). The StyleCLIP scores are poor, with high SFID and CSFID scores which was expected as StyleCLIP has been designed to work well where GANs shine. The StyleGAN2 model we use, trained

2. We used the publicly available model from <https://github.com/justinpinkney/awesome-pretrained-stylegan2>, and train our own e4e encoder (Tov et al. 2021) to embed images into this latent space.

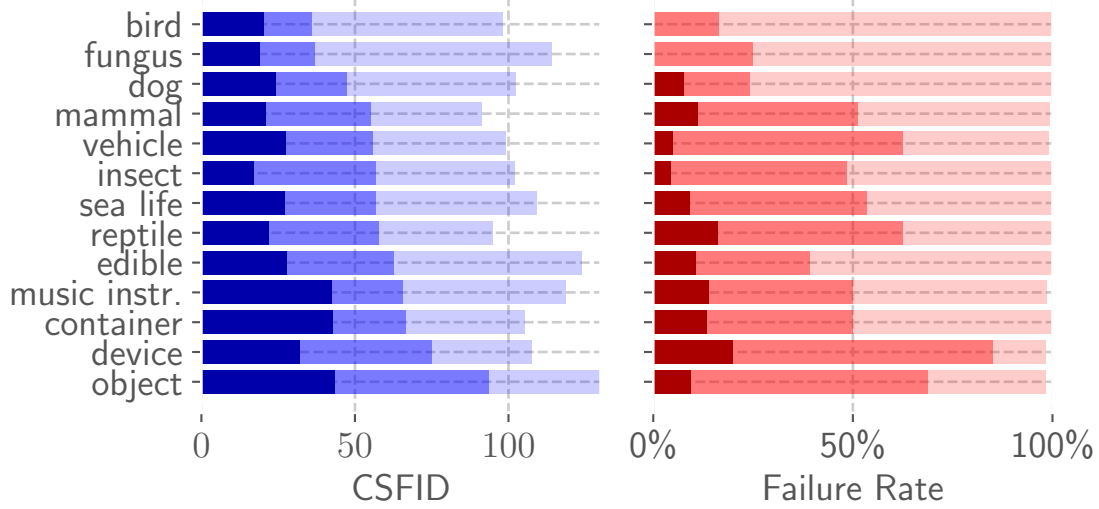


Figure 4.7. – Group-wise CSFID and Failure Rate (1-Accuracy), lower is better for both metrics. Dark colors: best possible values obtained with RETRIEVE baseline; medium colors: scores obtained with FLEXIT ; light colors: values obtained with COPY baseline.

on ImageNet, is agnostic to class information and cannot synthesize realistic images for all ImageNet classes. ManiGAN works well when trained on narrow domains with color change transformation requests, but we find that it does not produce convincing edits when trained on ImageNet.

To provide insight into which transformations work well, and which less so, we group our 47 ImageNet clusters into 13 bigger groups (see appendix for details) and report the average CSFID and failure rate (1-accuracy) scores for each group in Figure 4.7. Generally, transformations among natural objects are more successful than transformations among man-made objects. We believe that this is mostly because the latter appear in a wider variety of shapes and contexts which leads to more difficult transformations.

4.4.3 Ablation studies

Regularizers. In Figure 4.8, we show the evolution of CSFID along the optimization steps, where we consider our method without regularization, with each regularization scheme separately, and with all regularizers (default configuration). Compared to not using regularization, the LPIPS regularization substantially improves the CSFID score along the optimization path, while also reducing LPIPS as expected. The CLIP regularizer has a similar effect, but is able to reduce CSFID

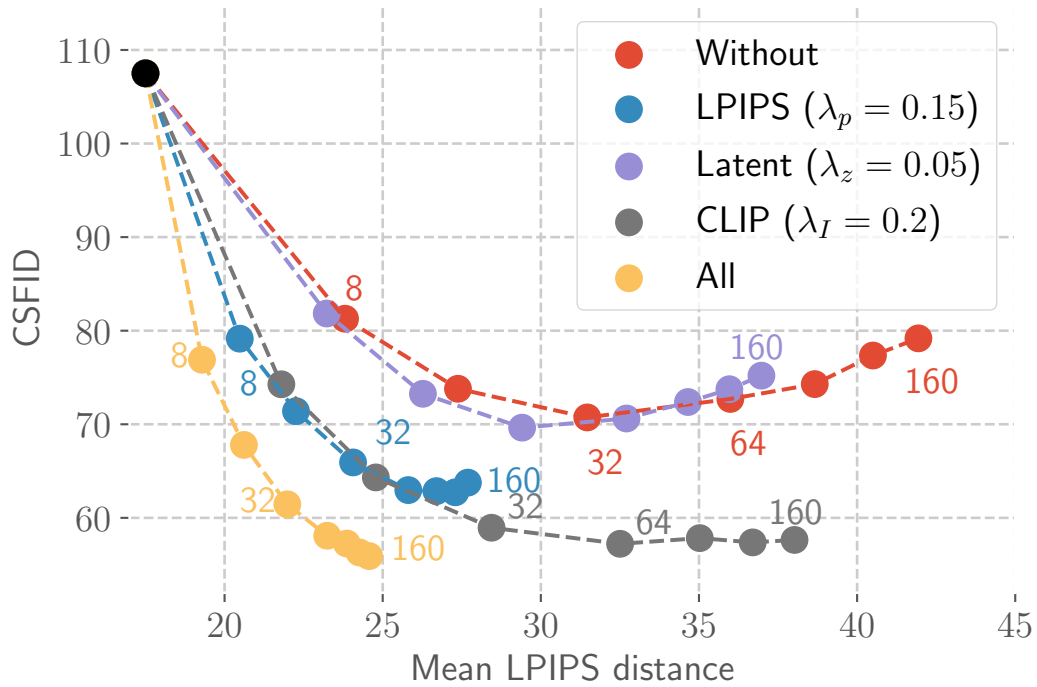


Figure 4.8. – CSFID obtained without regularization, with individual LPIPS, Latent and CLIP regularizers, and using all. Each curve corresponds to 160 steps of optimization on the dev. set.

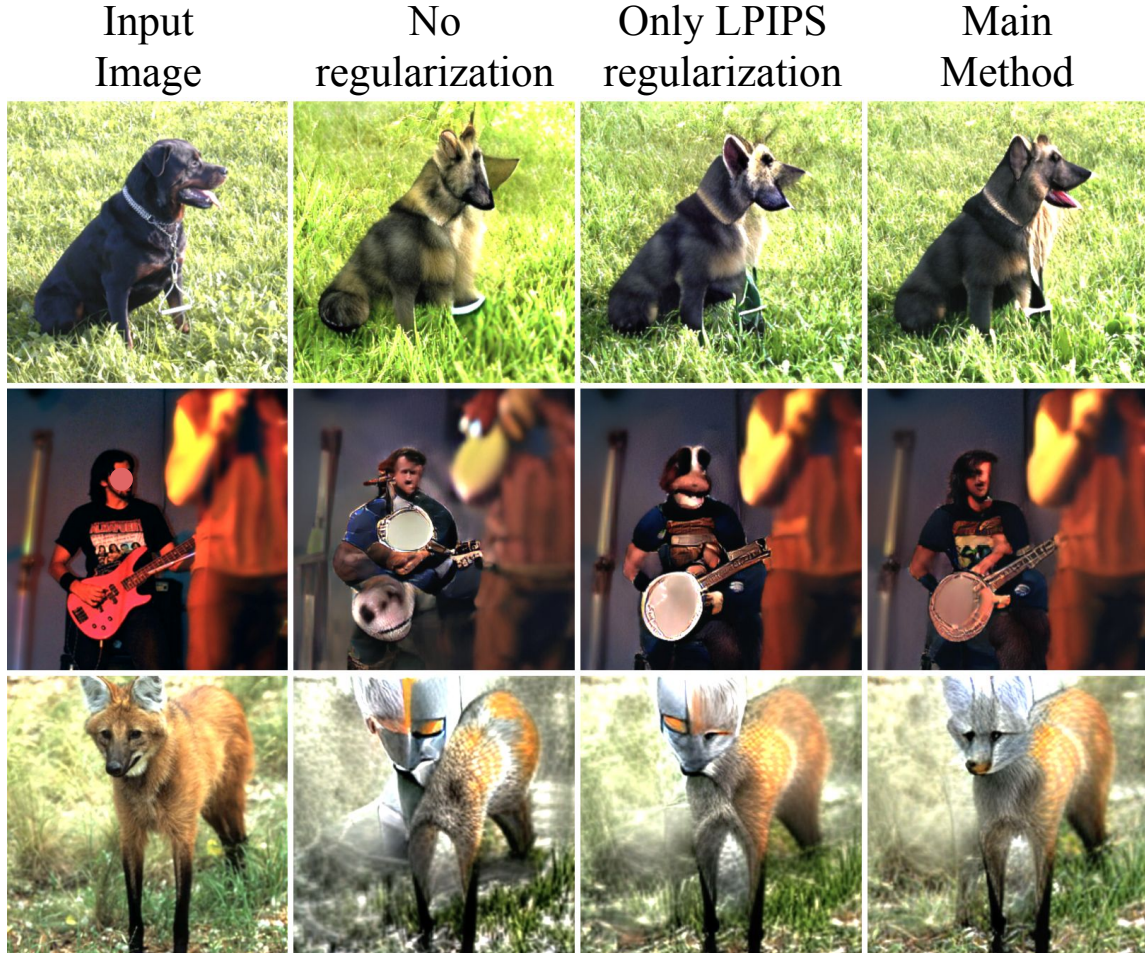


Figure 4.9. – Example transformations with different regularizers. Textual queries from top to bottom: Rottweiler → German shepherd, Electric guitar → Banjo, Red wolf → Grey fox.

further while the LPIPS distance is only slightly reduced compared to our method without any regularization. These two regularizers are complementary: while the LPIPS loss mitigates image deviation for local features, the CLIP loss provides semantic guidance which helps to reconstruct recognizable objects. Using all regularizers allows us to obtain the lowest CSFID scores at low LPIPS. Corresponding qualitative examples are shown in Figure 4.9.

CLIP embedding module. We study how different choices of CLIP image encoders impact the CSFID score. Our default configuration involves two ResNet-based networks and one ViT-based network to embed the image in the CLIP space. We experiment with a single ViT or ResNet, a combination of ViT with a single ResNet, and also using all available pre-trained CLIP networks, which comprises a ViT-B/16, a ViT-B/32, a ResNet50, ResNet50×4 and ResNet50×16, see (Radford

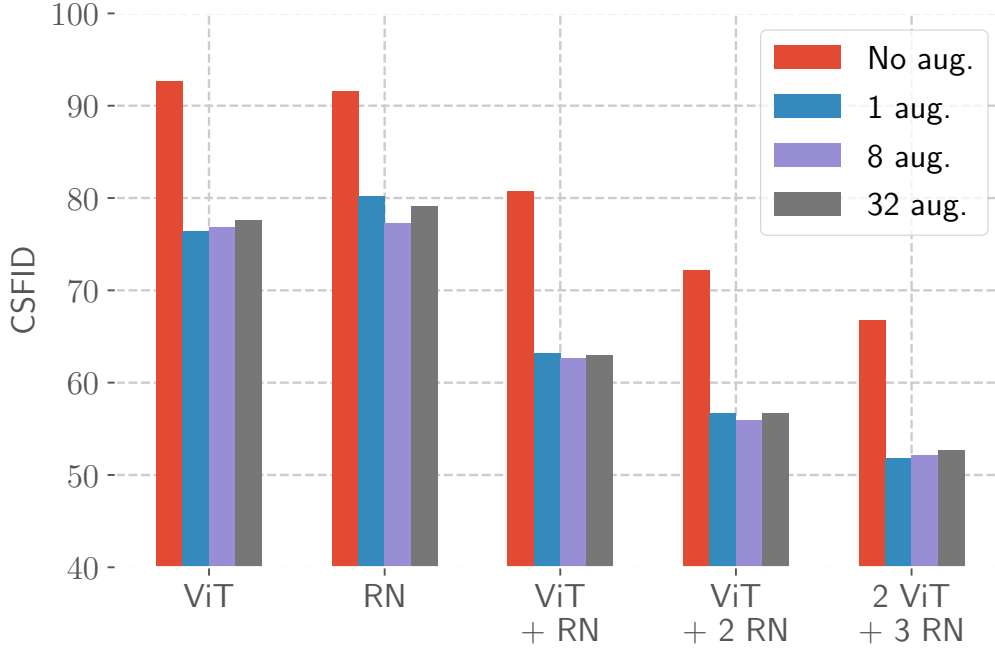


Figure 4.10. – CSFID for different CLIP networks combinations and number of data augmentations options. Default setting: ViT+2RN.

et al. 2021b) for details on the modules. For each CLIP network configuration, we experiment with either not using data augmentation, or using $d \in \{1, 8, 32\}$ augmentations, which are presented in Section 4.3. Each of the N_{nets} CLIP networks sees a different augmentation in each of the N_{steps} optimization steps, resulting in a total of $d \times N_{\text{nets}} \times N_{\text{steps}}$ augmentations of the input image.

From the results in Figure 4.10, we see that while the ViT and ResNet embedding networks lead to similar results, they are complementary and combining them leads to a substantial improvement. Adding additional networks leads to further improvements. Second, using data augmentation is very beneficial, and leads to a reduction in CSFID of 10 or more points for all network configurations. Using more than one augmentation does not improve results substantially: it suffices to a different augmentation for each network at each optimization step. In our other experiments we use the three smallest (and fastest) CLIP networks as our default setting.

Image optimization space. We compare our choice of optimizing in the VQ-GAN latent space with using the latent spaces of StyleGAN2 (T. Karras et al. 2020) and IC-GAN (Casanova et al. 2021), as well as optimizing directly in the pixel space. IC-GAN (Casanova et al. 2021) generates images similar to an input image, and uses a latent variable to allow for variability in its output. As IC-GAN does

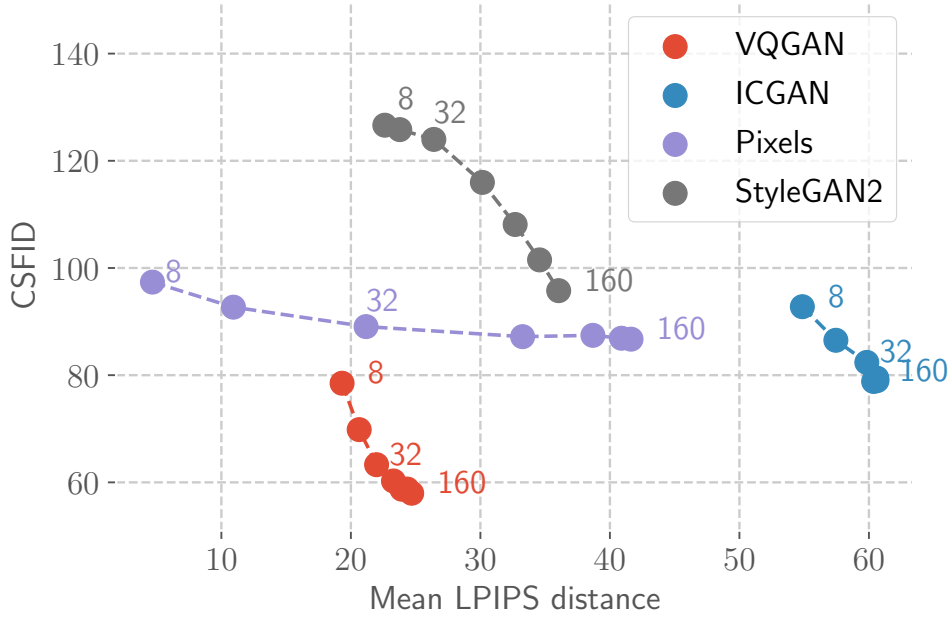


Figure 4.11. – CSFID and LPIPS scores across iterations, using different latent spaces, or raw pixels, for optimization.

not offer direct inference of the latents for a given image, we take 1,000 samples from the latent prior, and keep the one yielding minimal LPIPS distance to the input image. We found that optimization to further reduce the LPIPS w.r.t. the input image from this point on was not effective. For StyleGAN2 (T. Karras et al. 2020), we use the same network pretrained on ImageNet as we used for StyleCLIP. To embed the evaluation images into this latent space, we first obtain an initial prediction of the vector with the e4e encoder (Tov et al. 2021), as in StyleCLIP, and then perform an additional 1,000 optimization steps to better fit the input image, following the GAN inversion procedure described in (Tero Karras et al. 2019).

The results in Figure 4.11 show that using the VQGAN latent space allows to substantially decrease the CSFID score along the iterations, while only slightly increasing LPIPS. Using the raw pixel space is not effective to decrease the CSFID. IC-GAN has relatively good image synthesis abilities but it is hard to faithfully encode images in its latent space, yielding high LPIPS scores above 50. The StyleGAN2 latent space ($\mathcal{W}+$) is bigger, allowing generated images to be closer to the input images; however its CSFID scores are not competitive with the other approaches.

In Figure 4.12, we show qualitative results when we replace the VQGAN image encoder with other GAN-based encoders. VQGAN has a native encoder and decoder, and thus the initial latent vector is obtained directly. For StyleGAN2

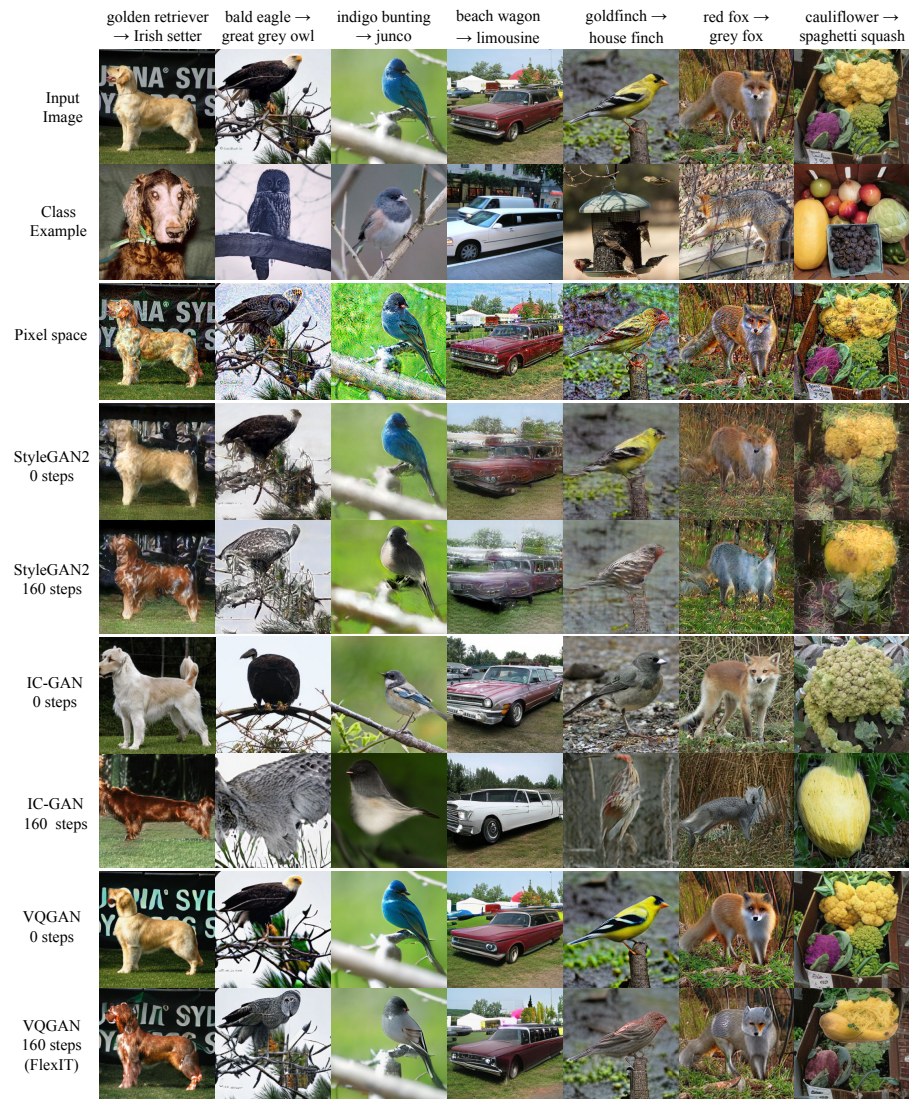


Figure 4.12. – Transformation examples for various image latent spaces.

T. Karras et al. 2020, we use the e4e encoder Tov et al. 2021 followed by an additional 1,000 steps of LPIPS minimization. For the IC-GAN Casanova et al. 2021 model, we use the BigGAN Brock et al. 2019 backbone as generator. IC-GAN is naturally conditioned on the SwaV embedding Caron et al. 2020 of the input image; for added robustness we sample 1,000 latent points and choose the one yielding smallest LPIPS distance with respect to the input image. For each latent space, we show the initial image decoded from the initial point z_0 , and the resulting image after 160 optimization steps. The three latent spaces differ substantially in their encoding images (0 steps). The IC-GAN latent space provides natural images that are far away from the input image due to the limited generator capacity in conjunction with the smaller latent space size (2560 dim.). StyleGAN2 images preserve the input image appearance thanks to the larger size of its latent space $\mathcal{W}+$ (8192), however images contain many unnatural artifacts due to the challenges of embedding images in this latent space Tov et al. 2021. The VQGAN latent space leads to the best reconstruction results. After 160 steps of optimization, the images generated with StyleGAN2 still have the same unnatural artifacts, and images generated with IC-GAN remain natural but far from the input images. VQGAN, which we use in FLEXIT, achieves good edits while preserving the overall image appearance. The pixel-space method introduces high-frequency artifacts, without substantially modifying the high-level semantic image content, resembling adversarial examples for image classification.

4.4.4 Hyperparameter study

In Figure 4.13, we illustrate the effect of our hyper-parameters on the LPIPS, CSFID, and Accuracy metrics. For the three regularization parameters λ_p , λ_z , λ_I , we observe that (i) the LPIPS distance with respect to the input image is smaller as the regularization gets stronger, as expected; (ii) less regularization allows more image modifications, yielding better accuracy scores, as illustrated in the bottom panel; (iii) there is a global minimum in CSFID scores when we vary each hyper-parameter independently (top panel). Regularization constraints are indeed useful to prevent inserting unnatural visual artifacts; however, too much regularization penalizes our algorithm as the distribution of output images gets closer to the input distribution, and thereby farther from the target distribution.

The parameter λ_S , similarly to the regularization parameters, has an optimal value which minimizes the CSFID. It is beneficial to give a hint to the optimization algorithm which semantic content should be changed, however focusing too much this objective reduces image realism.

For our main experiments, we set our hyper-parameters to minimize the CSFID score on the development set. This is a natural choice given the convex shape of

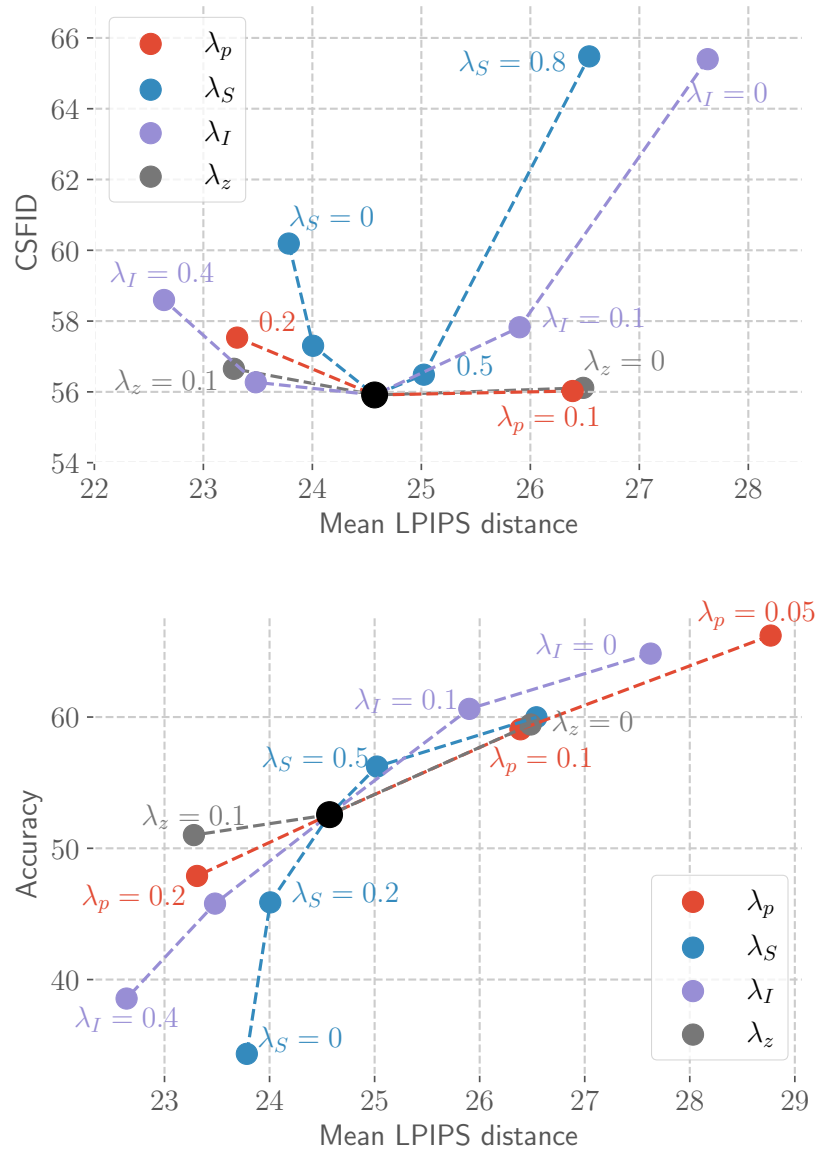


Figure 4.13. – Effect on CSFID and Accuracy of hyper-parameters; default settings represented by the black dot, where all lines cross.

the CSFID scores, whereas optimizing for accuracy would remove the regularizers which is detrimental for image quality.

4.4.5 Limitations

Figure 4.14 show representative failure cases for our method, due to either the regularization method or the multimodal embedding space. The first three columns show examples where the regularization with respect to the initial image was too strong.

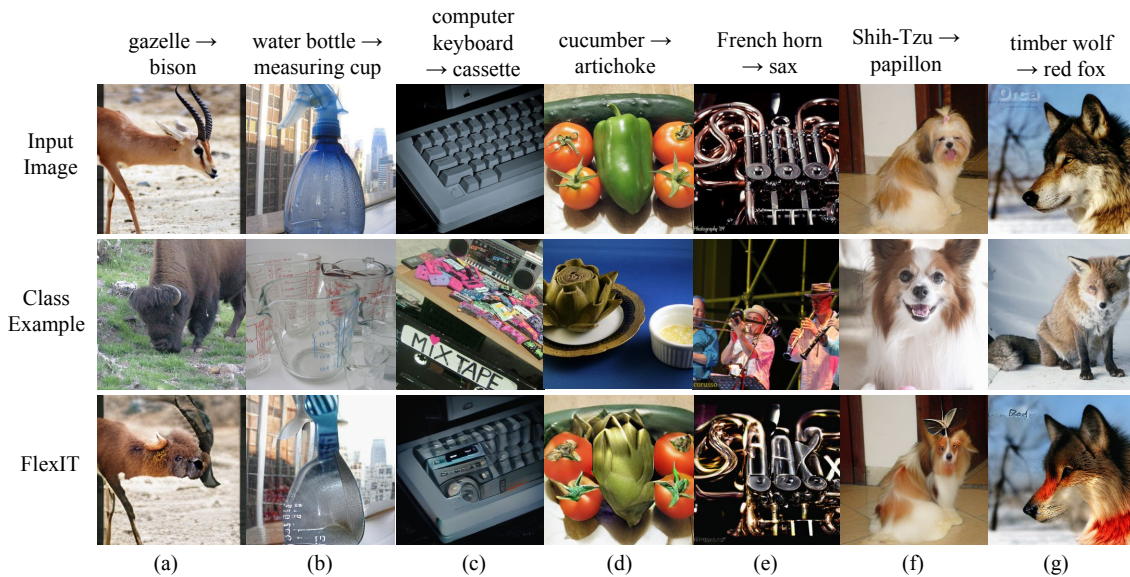


Figure 4.14. – Representative failure cases of FLEXIT.

- (a): FLEXIT added bison-like texture but fails to change the shape convincingly.
- (b): markings have been added to the bottle, but without changing its shape to that of a measuring cup.
- (c): only a part of the input object was changed.
- (d): the bell pepper rather than the cucumber was transformed, probably because the former is more centered, and has a better initial shape.

Columns (e)–(g) show failure cases related to the CLIP embedding space.

- (e): we observe an interesting text synthesis behavior where the letters of the target class “sax” have been written in the image. This is related to the OCR capabilities of CLIP.
- (f): a butterfly is synthesized on the head of the dog (CLIP optimized for both the dog breed papillon and the insect papillon).
- (g): an unrealistic image is produced by adding saturated red to the image.

4.5 Conclusion

In this chapter, we have presented FLEXIT, a novel method for semantic image translation. It uses a multimodal objective defined in the CLIP embedding space, and optimizes this objective with gradient descent the input image’s VQGAN latent representation. Using an autoencoder latent space, rather than specialized GAN latent spaces, lets us operate on a much wider range of images; using a general pretrained multi-modal embedding space provides flexibility, allowing FLEXIT to process free-text transformation queries without training.

While we studied transformations that change the class or color of the main object in a scene, other transformations of interest could consider changing the action of a subject (person walking v.s. running), changing object attributes, adding or deleting objects, or consider more elaborate textual descriptions which require non-trivial grounding in the image (“change the color of car parked next to the bicycle.”). Importantly, progress in this direction will require to identify the right data and evaluation metrics.

As our algorithm relies on CLIP for editing, it could potentially inherit its biases. The authors of CLIP have demonstrated that their model is subject to fairness issues such as misclassifying human faces into non-human or crime-related categories, and producing gender biased associations. Our editing method could reflect such biases if prompted transformations such as doctor \rightarrow newscaster, although we have not observed experimental evidence of this. A potential bias mitigation strategy would be to add constraints with CLIP prompts to control bias before and after editing.

Finally, FLEXIT works best for semantic image editing when the input image provides guidance, but has difficulties synthesizing realistic novel objects from scratch. The main reason is probably that despite our regularization schemes, the multimodal target embedding can sometimes be reached by synthesizing only characteristic parts of objects, the one which are best recognized by the CLIP network. This is because CLIP was trained with a contrastive classification

objective instead of a generative objective, and is therefore biased to recognize the most distinctive features in images and forget the less distinctive features.

One of the motivations for this work was that GANs could not be trained successfully for text-to-image tasks on wide distributions. Leveraging CLIP and its large-scale training allowed to process a much larger variety of images and edit prompts compared to standard GAN-based editing methods. However, instance-based optimization is much more computationally costly than GAN inference. On the other hand, recent diffusion models have been shown to scale very well with compute and data and are now able to generate photo-realistic images for a very large distribution of text prompts. In the next chapter, we propose to adapt these powerful models for semantic image editing.

IMAGE EDITING WITH DIFFUSION MODELS

5.1 Introduction

Text-conditional image generation is undergoing a revolution, with auto-regressive modelling and diffusion-based approaches (see Chapter 2.2) surpassing GANs on wide image distributions. Scaling these models is a key to their success: state-of-the-art models are now trained on vast amounts of data, which requires large computational resources (Chapter 2). Similarly to language models pretrained on web-scale data and adapted in downstream tasks with prompt engineering, the generative power of these big generative models can be harnessed to solve semantic image editing, avoiding training specialized architectures (B. Li et al. 2020a; J. Wang et al. 2022), or to use costly instance-based optimization (Crowson et al. 2022; Patashnik et al. 2021).

Diffusion models are an especially interesting class of model for image editing because of their iterative denoising process starting from random Gaussian noise. This process can be guided through a variety of techniques presented in Chapter 2, like CLIP guidance (Nichol et al. 2021; Avrahami et al. 2022b; Crowson 2021), and inpainting by copy-pasting pixel values outside a user-given mask (Lugmayr et al. 2022). These previous works, however, lack two crucial properties for semantic image editing: (i) inpainting discards information about the input image that should be used in image editing (e.g. changing a dog into a cat should not modify the animal’s color and pose); (ii) a mask must be provided as input to tell the diffusion model what parts of the image should be edited. We believe that while drawing masks is common on image editing tools like Photoshop, language-guided editing offers a more intuitive interface to modify images that requires less effort from users.

Conditioning a diffusion model on an input image can also be done without a mask, e.g. by considering the distance to input image as a loss function (Crowson 2021; J. Choi et al. 2021), or by using a noised version of the input image as a starting point for the denoising process as in SDEdit (Meng et al. 2021). However, these editing methods tend to modify the entire image, whereas we aim for

localized edits. Furthermore, adding noise to the input image discards important information, both inside the region that should be edited and outside.

In this chapter, we propose **DIFFEDIT**, a method that leverages a pretrained text-conditional diffusion model for zero-shot semantic image editing, without expensive editing-specific training. **DIFFEDIT** makes it possible by automatically finding what regions of an input image should be edited given a text query, by contrasting the predictions of a conditional and unconditional diffusion model. Edits obtained with **DIFFEDIT** are shown in Figure 5.1.

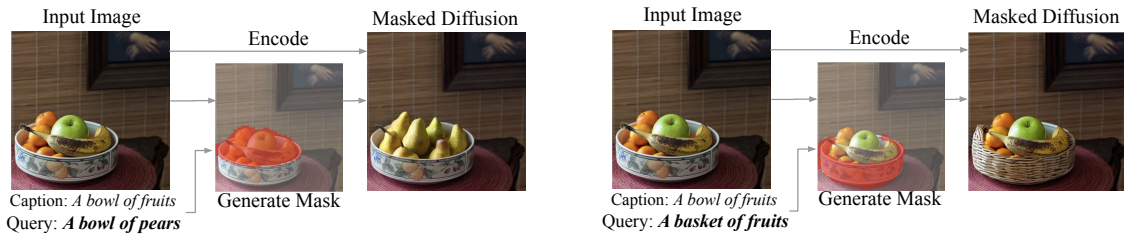


Figure 5.1. – In **DIFFEDIT**, a mask generation module determines which part of the image should be edited, and an encoder infers the latents, to provide inputs to a text-conditional diffusion model which produces the image edit.

We also show how using a reference text describing the input image and similar to the query, can help obtain better masks. Moreover, we demonstrate that using a reverse denoising model, to encode the input image in latent space, rather than simply adding noise to it, allows to better integrate the edited region into the background and produces more subtle and natural edits.

The remainder of this chapter is organized as follows: after going over related work, we present the **DIFFEDIT** framework along with a theoretical analysis showing how the distance between edited image and input image can be controlled. We then quantitatively evaluate our approach and compare to prior work using images of the ImageNet and COCO dataset, as well as a set of generated images.

5.2 Related work on Editing with Diffusion Models

Because diffusion models iteratively refine an image starting from random noise, they are easily adapted for inpainting when a mask is given as input, as exemplified in Chapter 2. J. Song et al. 2021 proposed to condition the generation process by copy-pasting pixel values from the reference image at each denoising step. Nichol et al. 2021 use a similar technique by copy-pasting pixels in the estimated final version of the image. T. Wang et al. 2022a use DDIM encoding of the

input image, and then decode on edited sketches or semantic segmentation maps. The gradient of a CLIP score can also be used to match a given text query inside a mask, as in Paint by Word (Bau et al. 2021), local CLIP-guided diffusion (Crowson 2021), or blended diffusion (Avrahami et al. 2022b). Lugmayr et al. 2022 apply a sequence of noise-denoise operations to better inpaint a specific region. There are also a number of methods that do not require an editing mask. In Diffusion-CLIP (G. Kim and Ye 2021), the weights of the diffusion model themselves are updated via gradient descent from a CLIP loss with a target text. The high computational cost of fine-tuning a diffusion model for each input image, however, makes it impractical as an interactive image editing tool. In SDEdit (Meng et al. 2021) the image is corrupted with Gaussian noise, and then the diffusion network is used to denoise it. While this method is originally designed to transform sketches to real images and to make pixel-based collages more realistic, we adapt it by denoising the image conditionally to the text query. In ILVR (J. Choi et al. 2021), the decoding process of diffusion model is guided with the constraint that downsampled versions of the input image and decoded image should stay close. Finally, in recent work concurrent to ours, Hertz et al. 2022 propose to edit images by modifying attention maps during the diffusion process.

5.3 DiffEdit algorithm

In this section, we first give an overview of diffusion models. We then describe our DIFFEDIT approach in detail, and provide a theoretical analysis comparing DIFFEDIT with SDEdit.

5.3.1 Background: diffusion models, DDIM and encoding

Denoising diffusion probabilistic models (Ho et al. 2020) is a class of generative models that are trained to invert a diffusion process. For a number of time steps \mathcal{T} , the diffusion process gradually adds noise to the input data, until the resulting distribution is (almost) Gaussian. A neural network is then trained to reverse that process, by minimizing the denoising objective

$$\mathcal{L} = \mathbb{E}_{\mathbf{x}_0, t, \epsilon} \|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|_2^2, \quad (5.1)$$

where ϵ_θ is the noise estimator which aims to find the noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ that is mixed with an input image \mathbf{x}_0 to yield $\mathbf{x}_t = \sqrt{\alpha_t} \mathbf{x}_0 + \sqrt{1 - \alpha_t} \epsilon$. The coefficient α_t defines the level of noise and is a decreasing function of the timestep t , with $\alpha_0 = 1$ (no noise) and $\alpha_{\mathcal{T}} \approx 0$ (almost pure noise).

J. Song et al. (2021) propose to use ϵ_θ to generate new images with the *DDIM* algorithm: starting from $\mathbf{x}_\mathcal{T} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the following update rule is applied iteratively until step 0:

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}} \left(\frac{\mathbf{x}_t - \sqrt{1 - \alpha_t} \epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{\alpha_t}} \right) + \sqrt{1 - \alpha_{t-1}} \epsilon_\theta(\mathbf{x}_t, t). \quad (5.2)$$

The variable \mathbf{x} is updated by taking small steps in the direction of ϵ_θ . Equation 5.2 can be written as the neural ODE, taking $\mathbf{u} = \mathbf{x}/\sqrt{\alpha}$ and $\tau = \sqrt{1/\alpha - 1}$:

$$d\mathbf{u} = \epsilon_\theta\left(\frac{\mathbf{u}}{\sqrt{1 + \tau^2}}, t\right) d\tau. \quad (5.3)$$

This allows to view DDIM sampling as an Euler scheme for solving Equation 5.3 with initial condition $\mathbf{u}(t = \mathcal{T}) \sim \mathcal{N}(\mathbf{0}, \alpha_\mathcal{T} \mathbf{I})$. This illustrates that we can use fewer sampling steps during inference than the value of \mathcal{T} chosen during training, by using a coarser discretization of the ODE. In the remainder of the chapter, we parameterize the time step t to be between 0 and 1, so that $t = 1$ corresponds to \mathcal{T} steps of diffusion in the original formulation. As proposed by J. Song et al. (2021), we can also use this ODE to encode an image \mathbf{x}_0 onto a latent variable \mathbf{x}_r for a time step $r \leq 1$, by using the boundary condition $\mathbf{u}(t = 0) = \mathbf{x}_0$ instead of $\mathbf{u}(t = 1)$, and applying an Euler scheme until time step r . In the remainder of the chapter, we refer to this encoding process as *DDIM encoding*, we denote the corresponding function that maps \mathbf{x}_0 to \mathbf{x}_r as E_r , and refer to the variable r as the *encoding ratio*. Similarly, we note D_r the inverse function that maps \mathbf{x}_r to \mathbf{x}_0 , which corresponds to regular DDIM decoding. With sufficiently small steps in the Euler scheme, decoding \mathbf{x}_r approximately recovers the original image \mathbf{x}_0 . This property is particularly interesting in the context of image editing: all the information of the input image \mathbf{x}_0 is encoded in \mathbf{x}_r , and can be accessed via DDIM sampling.

5.3.2 Semantic image editing with DiffEdit

In many cases, semantic image edits can be restricted to only a part of the image, leaving other parts unchanged. However, the input text query does not explicitly identify this region, and a naive method could allow for edits all over the image, risking modifying the input in areas where it is not needed. To circumvent this, we propose **DIFFEDIT**, a method to leverage a text-conditioned diffusion model to infer a mask of the region that needs to be edited. Starting from a DDIM encoding of the input image, **DIFFEDIT** uses the inferred mask to guide the denoising process, minimizing edits outside the region of interest. Figure 6.3 illustrates the three steps of our approach, which we detail below.

Step 1: Computing editing mask. When denoising an image, a text-conditioned diffusion model will yield different noise estimates given different text conditionings. We can consider *where* the estimates are different, which gives information about what image regions are concerned by the change in conditioning text. For instance, in Figure 6.3, the noise estimates conditioned to the query *zebra* and reference text *horse*¹ are different on the body of the animal, where they will tend to decode different colors and textures depending on the conditioning. For the background, on the other hand, there is little change in the noise estimates. The difference between the noise estimates can thus be used to infer a mask that identifies what parts on the image need to be changed to match the query. In our algorithm, we use a Gaussian noise with $t = 50\%$ (see analysis in Section 5.4.3, Figure 5.8), which means that the input image is linearly mixed with noise as $\mathbf{x} = \sqrt{\alpha_{0.5}}\mathbf{x}_0 + \sqrt{1 - \alpha_{0.5}}\epsilon$. We then estimate noise conditionally to the two different text conditionings, remove extreme values in noise predictions and stabilize the effect by averaging spatial differences over a set of n predictions, with $n = 10$ in our default configuration. The output is then rescaled to the range $[0, 1]$, and binarized with a threshold, which we set to 0.5 by default. The masks generally somewhat overshoot the region that requires editing, this is beneficial as it allows it to be smoothly embedded in its context, see examples in Section 5.4.

Step 2: Encoding. We encode the input image \mathbf{x}_0 in the implicit latent space at time step r with the DDIM encoding function E_r . This is done with the unconditional model, i.e. using conditioning text \emptyset , so no text input is used for this step.

Step 3: Decoding with mask guidance. After obtaining the latent \mathbf{x}_r , we decode it with our diffusion model conditioned on the editing text query Q , e.g. *zebra* in the example of Figure 6.3. We use our mask M to guide this diffusion process. Outside the mask M , the edited image should in principle be the same as the input image. We guide the diffusion model by replacing pixel values outside the mask with the latents \mathbf{x}_t inferred with DDIM encoding, which will naturally map back to the original pixels through decoding, unlike when using a noised version of \mathbf{x}_0 as typically done (Meng et al. 2021; J. Song et al. 2021). The mask-guided DDIM update can be written as $\tilde{\mathbf{y}}_t = M\mathbf{y}_t + (1 - M)\mathbf{x}_t$, where \mathbf{y}_t is computed from \mathbf{y}_{t-dt} with Equation 5.2, and \mathbf{x}_t is the corresponding DDIM encoded latent.

The encoding ratio r determines the strength of the edit: larger values of r allow for stronger edits that allow to better match the text query, at the cost of more deviation from the input image which might not be needed. We evaluate

1. We can also use an empty reference text, which we denote as $Q = \emptyset$.

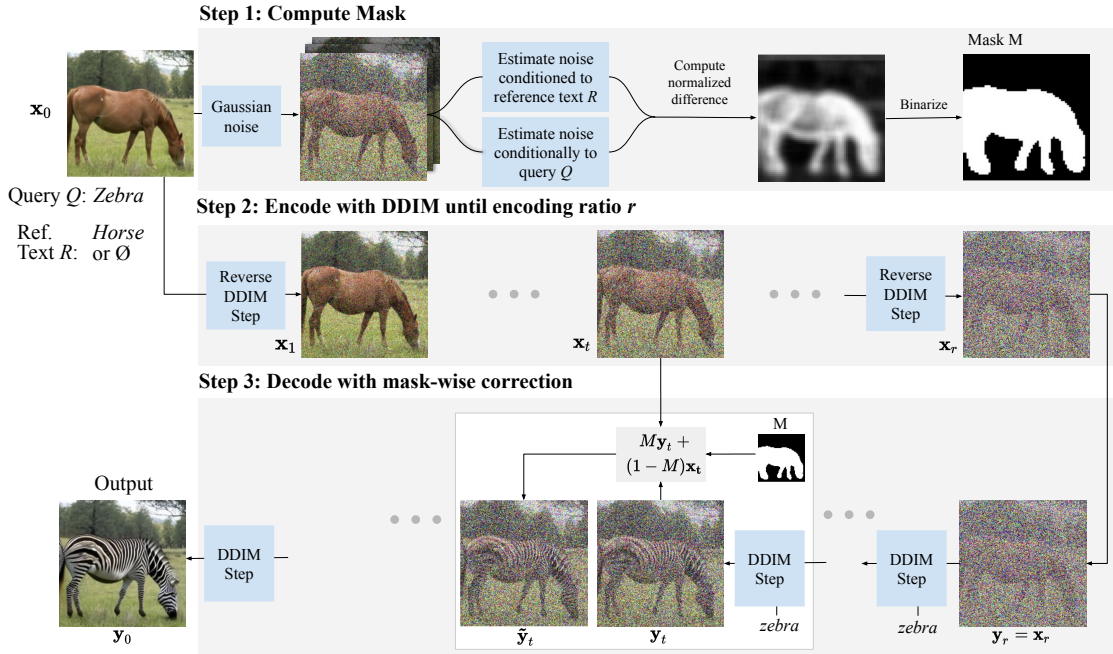


Figure 5.2. – The three steps of DIFFEDIT. **Step 1:** we add noise to the input image, and denoise it: once conditioned on the query text, and once conditioned on a reference text (or unconditionally). We derive a mask based on the difference in the denoising results. **Step 2:** we encode the input image with DDIM, to estimate the latents corresponding to the input image. **Step 3:** we perform DDIM decoding conditioned on the text query, using the inferred mask to replace the background with pixel values coming from the encoding process at the corresponding time step.

the impact of this parameter in our experiments. We illustrate the effect of the encoding ratio in figure 5.5.

5.3.3 Theoretical analysis

In DIFFEDIT, we use DDIM encoding to encode images before doing the actual editing step. In this section, we give theoretical insight on why this component yields better editing results than adding random noise as in SDEdit (Meng et al. 2021). With \mathbf{x}_r being the encoded version of \mathbf{x}_0 , using DDIM decoding on \mathbf{x}_r unconditionally would give back the original image \mathbf{x}_0 . In DIFFEDIT, we use DDIM decoding conditioned on the text query Q , but there is still a strong bias to stay close to the original image. This is because the unconditional and conditional noise estimator networks ϵ_θ and $\epsilon_\theta(\cdot, Q)$ often produce similar estimates, yielding similar decoding behavior when initialized with the same starting point \mathbf{x}_r . This means that the edited image will have a small distance w.r.t. the input image, a property critical in the context of image editing. We capture this phenomenon with the proposition below, where we compare the DDIM encoder $E_r(\mathbf{x}_0)$ to the SDEdit encoder $G_r(\mathbf{x}_0, \epsilon) := \sqrt{\alpha_r} \mathbf{x}_0 + \sqrt{1 - \alpha_r} \epsilon$, which simply adds noise to the image \mathbf{x}_0 .

Proposition 5.1. *Let $\mathcal{X} = \mathbb{R}^d$ be the space of input images, p_D be the data distribution of couples (\mathbf{x}_0, Q) where $\mathbf{x}_0 \in \mathcal{X}$ and Q a textual query to edit that image. Suppose that $\|\epsilon_\theta(\mathbf{x}_t, Q, t)\|_2 \leq C$ for all $\mathbf{x} \in \mathcal{X}$, $t \in [0, 1]$, that $\epsilon_\theta(\cdot, \emptyset, t)$ is K_1 -Lipschitz for all t , and let $K_2 = \mathbb{E}_{(\mathbf{x}_0, Q) \in p_D} \max_{t \in [0, 1]} \|\epsilon_\theta(\mathbf{x}, Q, t) - \epsilon_\theta(\mathbf{x}, \emptyset, t)\|$. Then, for all encoding ratios $0 \leq r \leq 1$, we have the two following bounds:*

$$\mathbb{E}_{\substack{(\mathbf{x}_0, Q) \sim p_D \\ \epsilon \sim \mathcal{N}(0, 1)}} \|\mathbf{x}_0 - D_r(G_r(\mathbf{x}_0, \epsilon), Q)\|_2 \leq (C + 1)\tau, \quad (5.4)$$

$$\mathbb{E}_{(\mathbf{x}_0, Q) \sim p_D} \|\mathbf{x}_0 - D_r(E_r(\mathbf{x}_0), Q)\|_2 \leq \frac{K_2 \tau}{\sqrt{\tau^2 + 1}} \left(\tau + \sqrt{\tau^2 + 1} \right)^{K_1}, \quad (5.5)$$

where $\tau = \sqrt{1/\alpha_r - 1}$ increases with the encoding ratio r : $\tau(r = 0) = 0$ and $\lim_{r \rightarrow 1} \tau = +\infty$.

We provide the proof in Appendix A.4. The first bound is associated with SDEdit, and is an extension of a bound proven in the original paper. The second bound we contribute is associated with DIFFEDIT. It is tighter than the first bound below a certain encoding ratio, see Figure 5.3. We empirically estimated the parameters K_1 , K_2 and C with the diffusion models that we are using. While the asymptotic behavior of the second bound is worse than the first with $K_1 > 1$, it is the very small value of K_2 that gives a tighter bound.

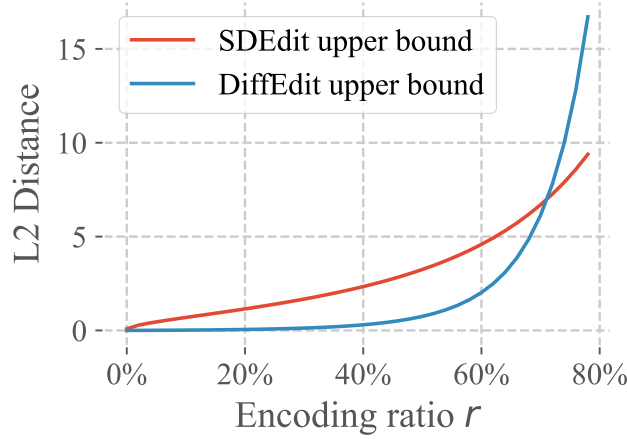


Figure 5.3. – Illustration of the bounds from Proposition 5.1, with estimated parameters $C=1$, $K_2=0.02$, and $K_1=3$.

This supports our argument from above: because the unconditional and text-conditional noise estimates generally give close results — K_2 being a measure of the average difference— the Euler scheme with $\epsilon_\theta(\cdot, Q, \cdot)$ gives a sequence of intermediate latents $\mathbf{y}_r, \dots, \mathbf{y}_0$ that stays close to the trajectory $x_r, \dots, D_r(x_r) \approx x_0$ mapping back x_r to x_0 . While these upper bounds do not guarantee that DDIM encoding yields smaller edits than SDEdit, experimentally we find that it is indeed the case.

5.4 Experiments

In this section, we describe our experimental setup, followed by qualitative and quantitative results.

5.4.1 Experimental setup

Datasets. We perform experiments on three datasets. First, on *ImageNet* (Jia Deng et al. 2009) we follow the evaluation protocol of FlexIT (Couairon et al. 2022b). Given an image belonging to one class, the goal is to edit it so that it will depict an object of another class as indicated by the query. Given the nature of the ImageNet dataset, edits often concern the main object in the scene. Second, we consider editing images generated by *Imagen* (Saharia et al. 2022b) based on structured text prompts, in order to evaluate edits that involve changing the background, replacing secondary objects, or changing object properties. Third, we

consider edits based on images and queries from the COCO (T.-Y. Lin et al. 2014a) dataset to evaluate edits based on more complex text prompts.

Diffusion models. In our experiments we use latent diffusion models (Rombach et al. 2022b). We use the class-conditional model trained on ImageNet at resolution 256×256 , as well as the 890M parameter text-conditional model trained on LAION-5B (Schuhmann et al. 2021), known as *Stable Diffusion*, at 512×512 resolution.² Since these models operate in a VQGAN latent spaces (Esser et al. 2021b), the resolution of our masks is 32×32 (ImageNet) or 64×64 (Imagen and COCO). We use 50 steps in DDIM sampling with a fixed schedule, and the encoding ratio parameter further decreases the number of updates used for our edits. This allows to edit images in ~ 10 seconds on a single Quadro GP100 GPU. We also use classifier-free guidance (Ho and Salimans 2022) with the recommended values: 5 on ImageNet, 7.5 for Stable Diffusion.

Comparison to other methods. We use SDEdit (Meng et al. 2021) as our main point of comparison, since we can use the same diffusion model as for DIFFEDIT. We also compare to FlexIT (Couairon et al. 2022b), a mask-free, optimization-based editing method based on VQGAN and CLIP. On ImageNet, we evaluate ILVR (J. Choi et al. 2021) which uses another diffusion model trained on ImageNet (Dhariwal and Nichol 2021b). Finally, on COCO and Imagen images, we compare to the concurrent work of Hertz et al. 2022.³

Evaluation. In semantic image editing, we have to satisfy the two contradictory objectives of (i) matching the text query and (ii) staying close to the input image. For a given editing method, better matching the text query comes at the cost of increased distance to the input image. Different editing methods often have a parameter that allows to control the editing strength: varying its value allows to get different operating points, forming a trade-off curve between the two objectives aforementioned. Therefore, we evaluate editing methods by comparing their trade-off curves. For diffusion-based methods, we use the encoding ratio to control the trade-off.

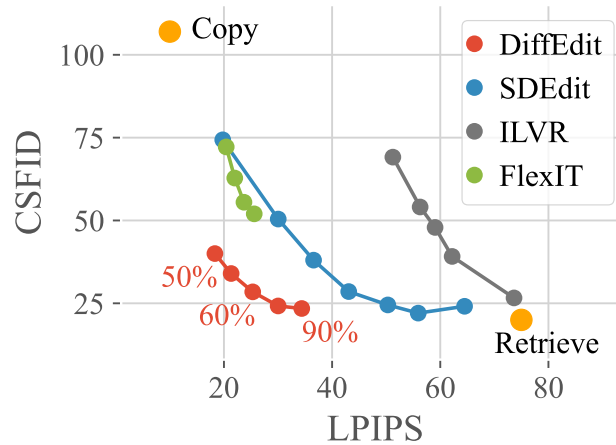


Figure 5.4. – Comparison on ImageNet data of DIFFEDIT with other Image Editing methods. For DIFFEDIT we annotate the different operating points with the corresponding encoding ratios.

5.4.2 Experiments on ImageNet

On ImageNet, we follow the evaluation protocol of Couairon et al. 2022b, with the associated metrics: the LPIPS perceptual distance (R. Zhang et al. 2018) measures the distance with the input image, and the CSFID, which is a class-conditional FID metric (Heusel et al. 2017b) measuring both image realism and consistency w.r.t. the transformation prompt. For both metrics, lower values indicate better edits. For more details see Couairon et al. 2022b.

We compare DIFFEDIT to other semantic editing methods from the literature in terms of CSFID-LPIPS trade-off. Stronger edits improve (lower) the CSFID score as the edited images better adhere to the text query, but the resulting images tend to deviate more from the input image, leading to worse (increased) LPIPS distances.

The results in Figure 5.4 indicate that DIFFEDIT obtains the best trade-offs among the different methods. For fair comparison with previous methods, here we do not leverage the label of the input image and use the empty text as reference when inferring the editing mask. The *Copy* and *Retrieve* baselines are two opposite cases where we have the best possible LPIPS distance —zero, by copying the input image— and best possible transformation score by discarding the input image and replacing it with a real image from the target class from the ImageNet

2. Available at <https://huggingface.co/CompVis/stable-diffusion>.

3. As there is no official implementation available at the time of writing, we used the unofficial implementation adapted for Stable Diffusion from <https://github.com/bloc97/CrossAttentionControl>.

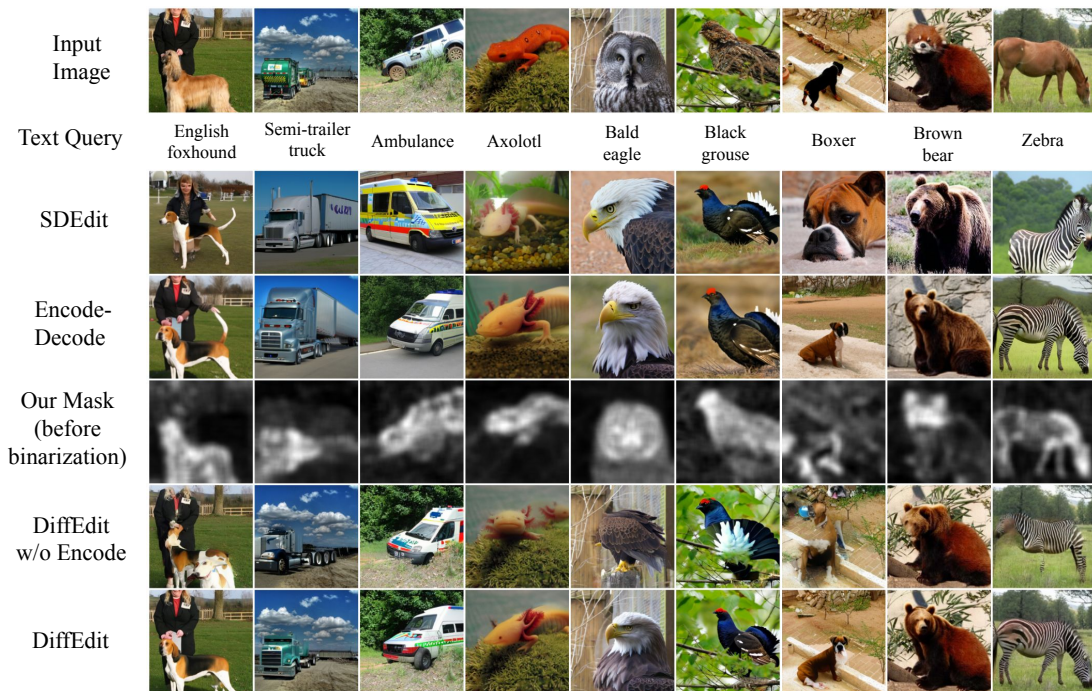


Figure 5.5. – Edits obtained on ImageNet with `DIFFEDIT` and ablated models. Encode-Decode is `DIFFEDIT` without masking, and `SDEdit` is obtained when not using masking nor encoding. When not using masking (`SDEdit` and Encode-Decode) we observe undesired edits to the background, see e.g. the sky in the second column. When not using DDIM encoding (`SDEdit` and `DIFFEDIT` w/o Encode), appearance information from the input —such as pose— is lost, see last two columns.

dataset. **DIFFEDIT**, as well as the diffusion-based **SDEdit** and **ILVR**, are able to obtain CSFID values comparable to that of the retrieval baseline. Among the diffusion-based methods, our **DIFFEDIT** obtains comparable CSFID values at significantly better LPIPS scores. For **FlexIT**, the best CSFID value is significantly worse, indicating it is not able to produce both strong and realistic edits. Using more optimization steps does not solve this issue, as the distance to the input image is part of the loss it minimizes.

Visual ablation. We show visual results for ablations of our two main components, mask inference and DDIM encoding, in Figure 5.6. The resulting methods are **SDEdit** (Meng et al. 2021), Encode-Decode, **DIFFEDIT** w/o Encoding, and **DIFFEDIT**. We demonstrate the qualitative behavior of these different methods, at varying encoding ratios between 30% and 80%. Compared to **SDEdit**, Encode-Decode allows to better match the query with fewer modifications of the main object and the background, especially at 60% – 70%. Mask inference allows to maintain exactly the background. Using DDIM inference on top of mask-based decoding allows to better retain visual appearance inside the mask, especially at 70% and 80%, c.f. row 3 vs. 4.

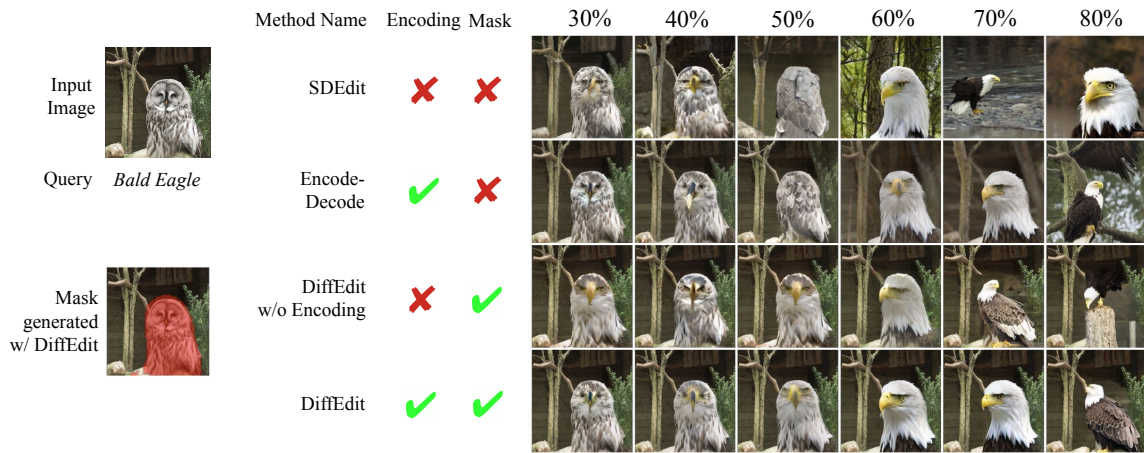


Figure 5.6. – Qualitative ablations of the mask and encoding components, using different encoding ratios from 30% to 80%.

Ablation experiments. We ablate the two core components of **DIFFEDIT**, mask inference and DDIM encoding, to measure their relative contributions in terms of CSFID-LPIPS trade-off. If we do not use either of these components our method reverts to **SDEdit**. The results in Figure 5.7, left panel, show that adding DDIM encoding (Encode-Decode) and the masking (**DIFFEDIT** w/o Encode) separately both improve the trade-off and reduce the average editing distance w.r.t. the

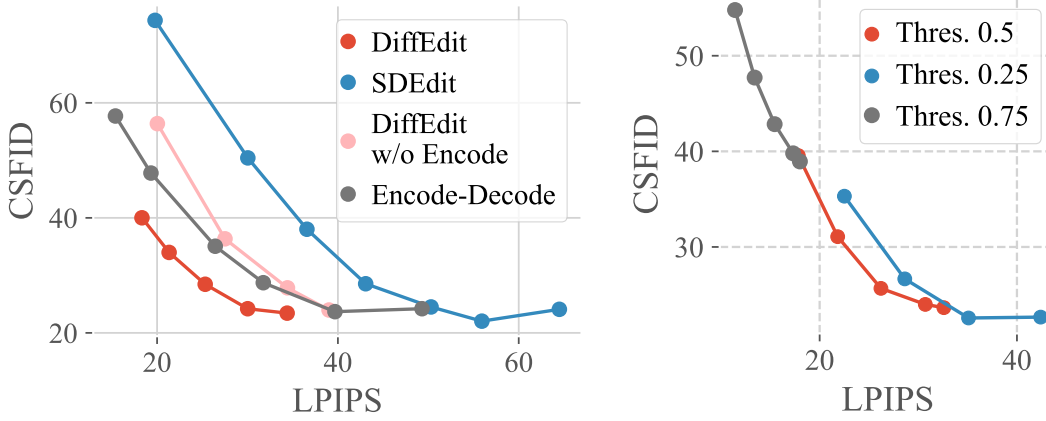


Figure 5.7. – Ablations on ImageNet. Left: effect of masking and encoding component. Right: DIFFEDIT with different mask thresholds; with 0.5 our default setting.

input image compared to SDEdit. Moreover, combining these two elements into DIFFEDIT gives an even better trade-off, showing their complementarity.

The right panel of Figure 5.7 shows DIFFEDIT with different mask binarization thresholds. Compared to our default value of 0.5, a lower threshold of 0.25 results in larger masks (more image modifications) and worse CSFID-LPIPS trade-off. A higher threshold of 0.75 results in masks that are too restrictive: the CSFID score stagnates around 40, even at large encoding ratios.

Finally, our mask guidance operator $\tilde{y}_t = M\mathbf{y}_t + (1 - M)\mathbf{x}_t$ provides a better trade-off than the operator used in GLIDE (Nichol et al. 2021), which interpolates \mathbf{y}_t with a mask-corrected version of the predicted denoised image $\hat{\mathbf{y}}_0$. With encoding ratio 80%, both operators produce edits with a LPIPS score of 30.5, but the GLIDE version yields a CSFID of 26.4 compared to 23.6 for ours.

5.4.3 Analysis of noise used to compute the mask

In step one our method an editing mask is inferred by contrasting noise estimations on a *noised* version of the input image, see Section 5.3.2. In this section, we study the impact of the level of noise added to the input image, by varying its value between 0.1 and 0.8, where 0 corresponds to using the initial image as input, and 1 to replacing the input image with random Gaussian noise. We evaluate the obtained operating points on ImageNet with the CSFID and LPIPS metrics when using an encoding ratios of 0.7 and 0.8 for DDIM encoding and masked-guided denoising in steps two and three of DIFFEDIT. From the results in Figure 5.8,

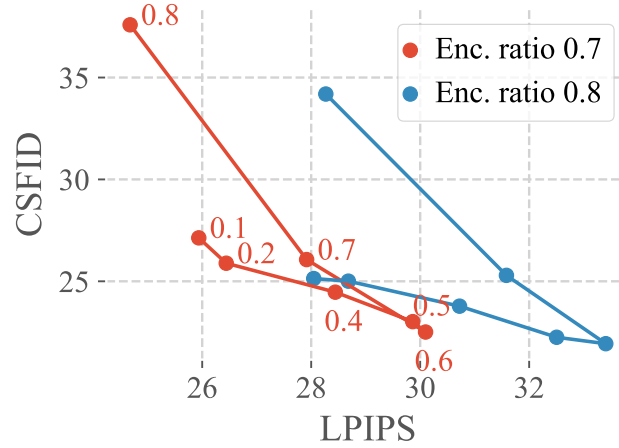


Figure 5.8. – Impact of the noise added to input image when computing the mask, for encoding ratios of 0.7 and 0.8 on ImageNet.

we find that best results are obtained for moderate values of noise addition of 0.6 and below. Indeed, with too much noise added to the input image, it is difficult to correctly identify visual elements in the input image. We use a value of 0.5 in all our experiments.

5.4.4 Experiments on images generated by Imagen

In our second set of experiments we evaluate edits that involve changes in background, replacing secondary objects, and editing object properties. We find that images generated by Imagen (Saharia et al. 2022b) offer a well suited test bed for this purpose. Indeed, the authors tested the compositional abilities of Imagen with template prompts of the form: “{A photo of a | An oil painting of a} {fuzzy panda | British shorthair cat | Persian cat | Shiba Inu dog | raccoon} {wearing a cowboy hat and | wearing sunglasses and} {red shirt | black jacket} {playing a guitar | riding a bike | skateboarding} {in a garden | on a beach | on top of a mountain}”, resulting in 300 prompts.

We use the generated images as input and ask to change the prompt to another prompt for which one of these elements is changed. Since we cannot use the CSFID metric as for ImageNet, as images do not carry a single class label, we use FID to measure image realism, and CLIPScore (Hessel et al. 2021) to measure the alignment of the query and output image. These two scores have become the standard in evaluating text-conditional image generation (Saharia et al. 2022b).

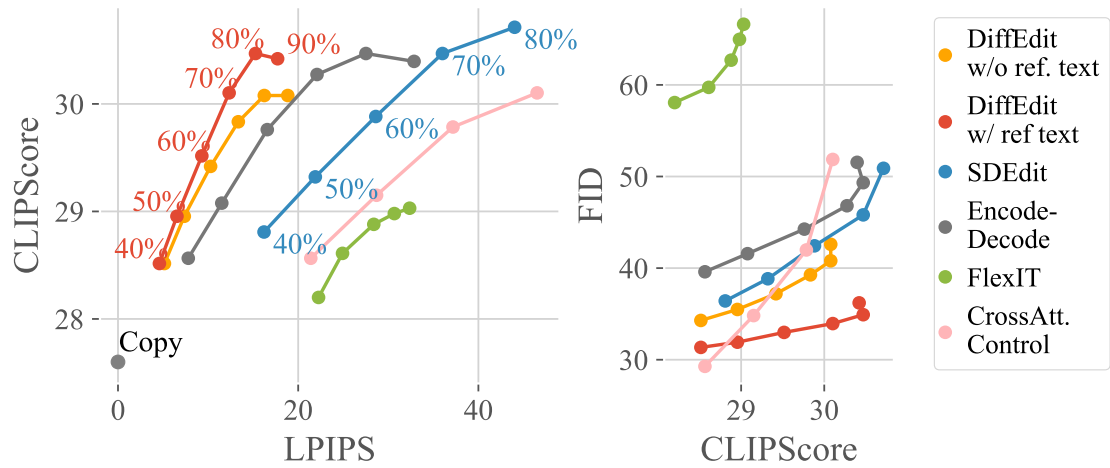


Figure 5.9. – Editing trade-offs on Imagen images.

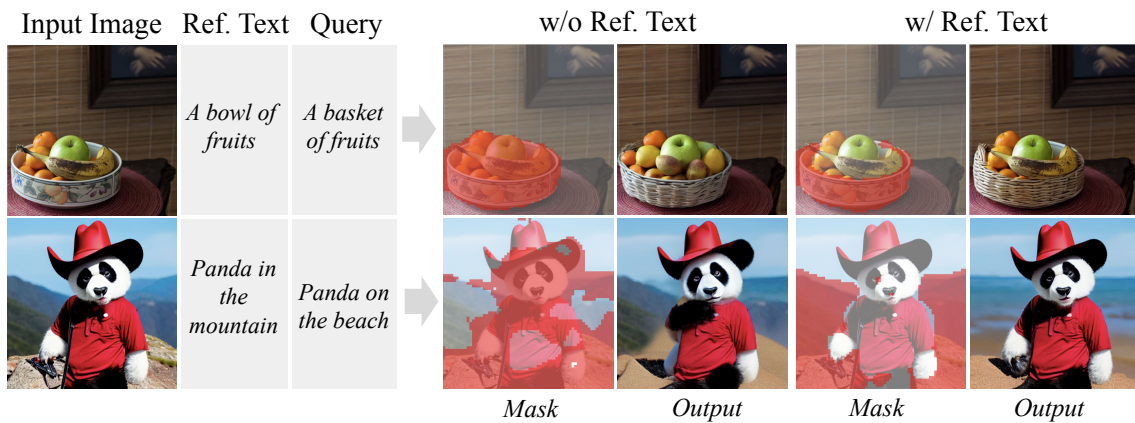


Figure 5.10. – Masks and edits obtained with and without reference text in the mask computation algorithm.

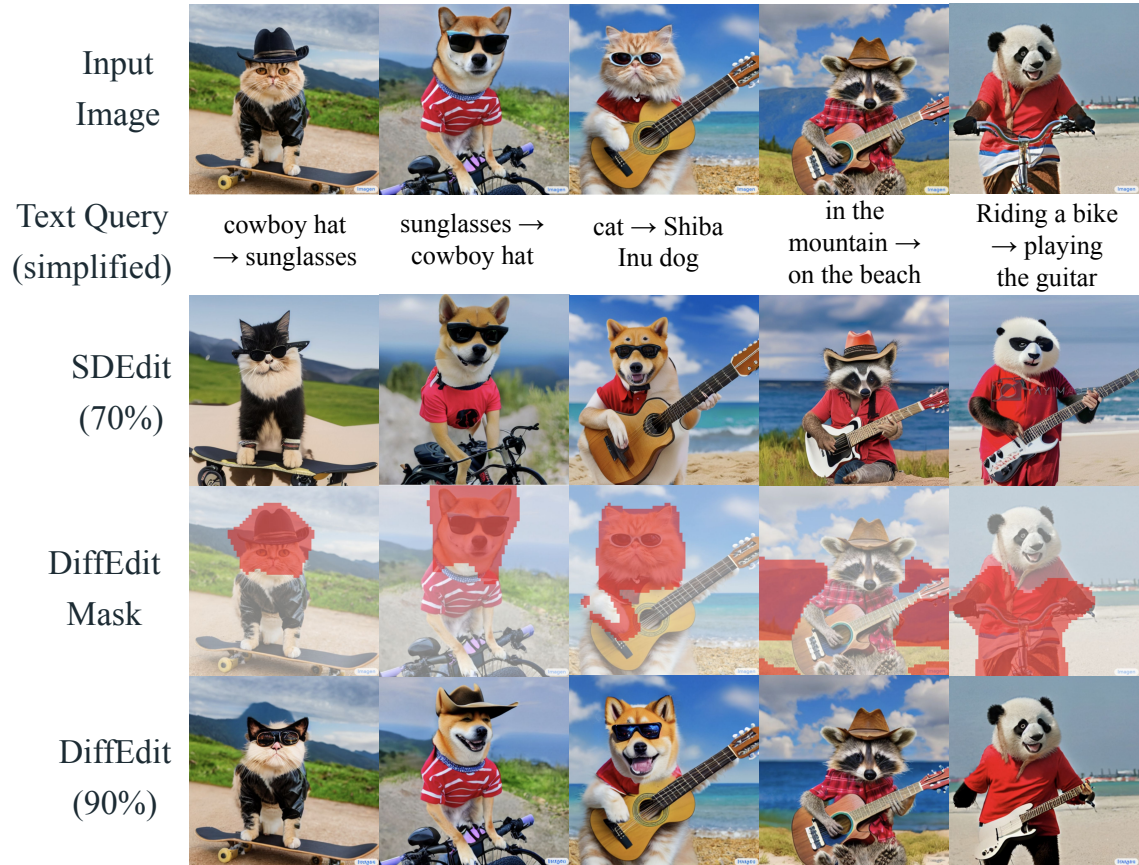


Figure 5.11. – Edits on Imagen dataset. We use encoding ratio of 90% for DIFFEDIT and 70% for SDEdit for fair comparison: both methods have similar CLIPScore, for larger encoding ratios SDEdit drastically change the input.

Figure 5.9 displays the CLIP-LPIPS and FID-CLIP trade-offs. DIFFEDIT provides more accurate edits than SDEdit, FlexIT, and Cross Attention Control, by combining inferred masks with DDIM encoding. Two versions of DIFFEDIT are shown, which differ by how the mask is computed: they correspond to (i) using the original caption as reference text (labelled *w/ ref. text*) or (ii) using the empty text \emptyset (labelled *w/o ref. text*).

Computing the mask with the original caption as reference text yields the best overall trade-off. Leveraging the original caption yields better CLIP and FID scores. Figure 5.10 illustrates the difference in the masks obtained with and without reference text for two examples. The reference text allows ignoring parts of the image that are described both by the query and reference text (e.g. the fruits), because in both cases the network uses the common text on the corresponding image region to estimate the noise. On the contrary, parts where the query and reference text disagree, e.g. “*bowl*” vs. “*basket*”, will have different noise estimates. Qualitative transformation examples are shown in Figure 5.11, where the masks are inferred by contrasting the caption and query texts.

5.4.5 Experiments on COCO

To evaluate semantic image editing with more complex prompts, we use images and captions from the COCO dataset T.-Y. Lin et al. 2014a. To this end, we leverage the annotations provided by Hexiang Hu et al. 2019, which associate images from the COCO validation set with other COCO captions that are similar to the original ones, but in contradiction with the given image. This makes these annotations particularly interesting as queries for semantic image editing, as they can often be satisfied by editing only a part of the input image, see Figure 5.12 for examples. Similar to our evaluation for Imagen images, here we evaluate edits in terms of CLIPScore, FID and LPIPS.

The results in Figure 5.13 show that the CLIP-LPIPS trade-off of DIFFEDIT is the best, but that it reaches lower maximum CLIP score than SDEdit. The FID scores are similar to SDEdit, but significantly improves upon the Encode-Decode ablation, which does not use a mask.

Moreover, in contrast to results on the Imagen data, leveraging the original image caption does not change the CLIP-LPIPS and FID-CLIP trade-offs. We find that the caption often describes the input image differently compared to the query text, making it more difficult to identify which part of the image needs to be edited. We verify this hypothesis in Section A.3 by filtering the dataset according to the edit distance between the caption and edit query. When the caption and




Dataset	COCO	COCO	COCO
Input Image			
Caption	A modern style bathroom with a large tub and shower and tile floor.	A large tall tower with a clock on top.	Three giraffe stand near a fence as two women watch them.
Text Query	A bathroom with sheer curtains framing the tub.	A tall clock tower with trees all around.	Two giraffe standing next to each other near brick building.

Figure 5.12. – Editing queries on the COCO dataset.

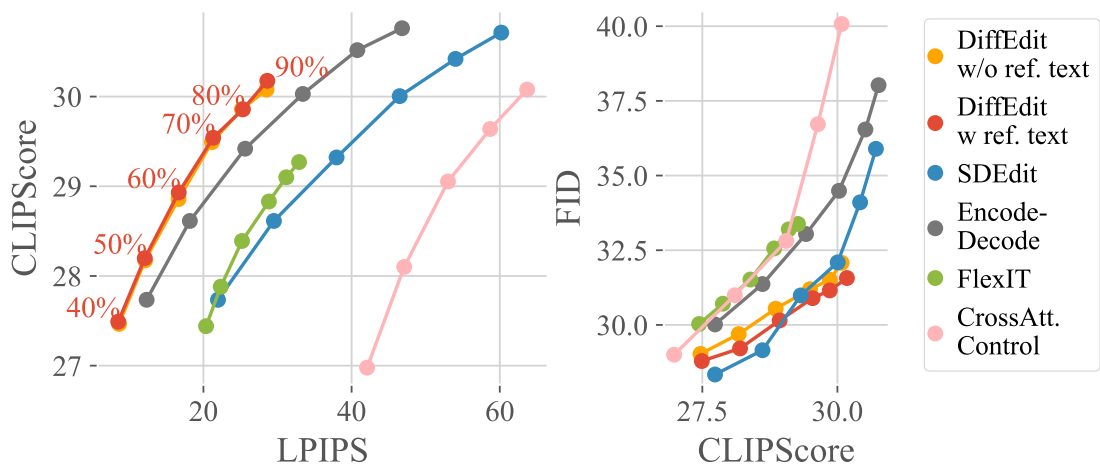


Figure 5.13. – Quantitative evaluation on COCO.

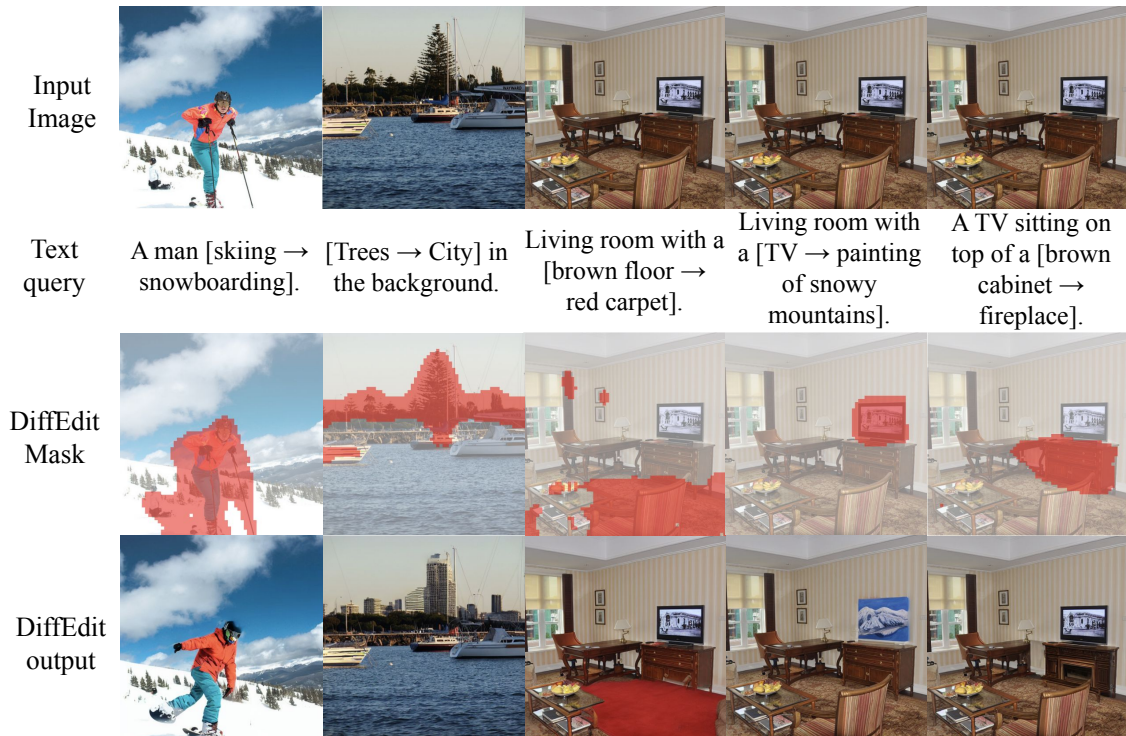


Figure 5.14. – Examples edits on COCO images.

edit query are similar, leveraging the image caption boosts CLIP scores by 0.25 points, a similar improvement as seen on the Imagen data.

Qualitative examples are shown in Figure 5.14. The first column illustrates the benefit of DDIM encoding: we are able to correctly maintain properties of the object inside the mask, such as clothes’ color. The three last columns illustrate how contrasting different pairs of reference and query text allows selecting different objects in the input image to perform different edits.

Additional qualitative examples on COCO images are shown in Figure 5.15 and Figure 5.16.

5.4.6 Representative failure cases

Figure 5.17 shows several failure cases of semantic image editing with DIFFEDIT. Some failure modes are inherited from the generative model itself: models trained on web-scraped image-text data are known to struggle with understanding spatial positions in images, spatial reasoning, and counting (Ramesh et al. 2021). Others are specific to our mask-based method, like the difficulty to insert objects, because the mask often seeks an “anchor” visual element to insert an object, see first column.

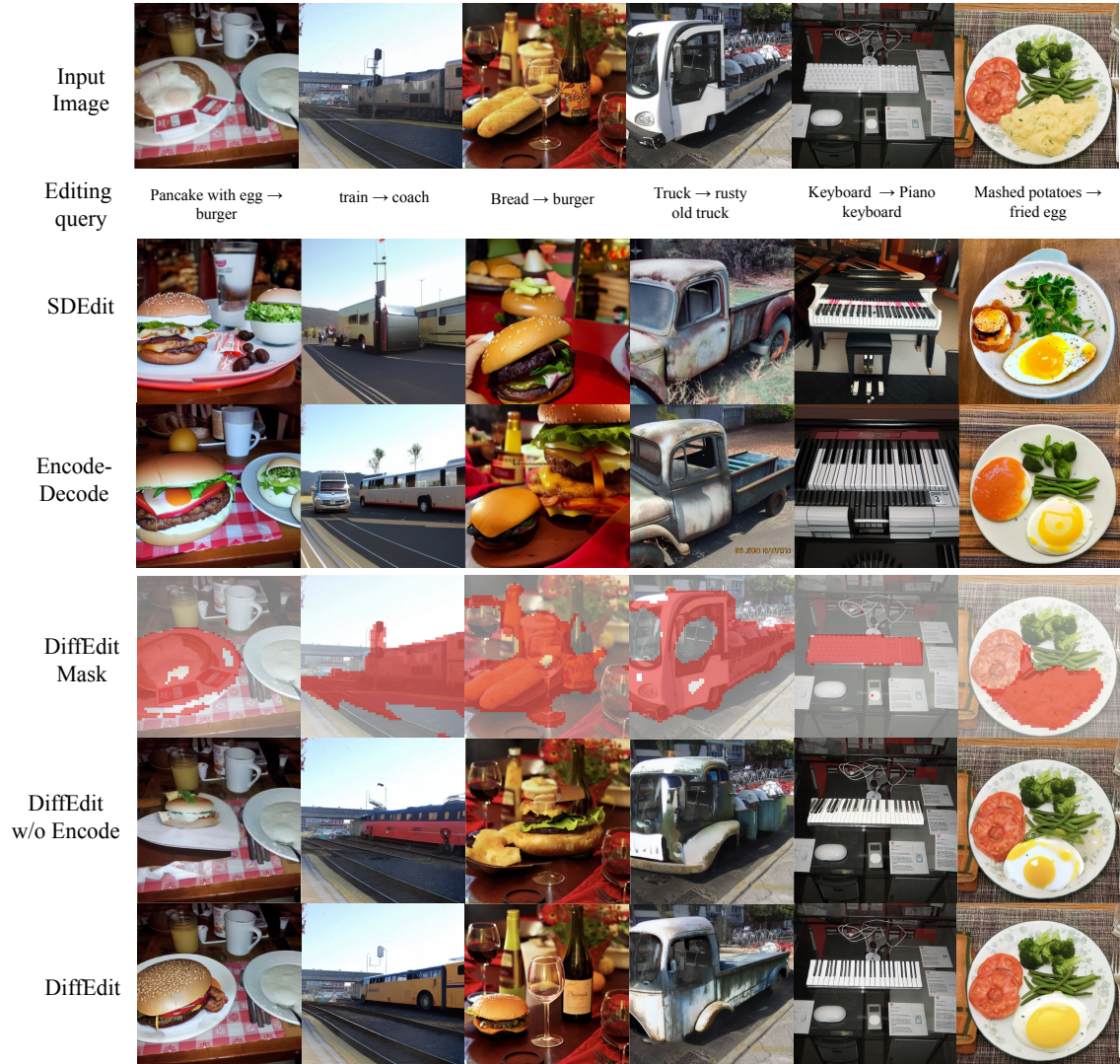


Figure 5.15. – More qualitative examples on COCO. Baseline methods are shown for comparison. The mask is sometimes bigger or smaller than one could expect: in column 3, it is larger, but there are few edits outside the requested *bread* → *burger* transformation (except for the wine bottle label), which is not the case without DDIM encoding. In column 4, the mask does not cover the interior of the truck, but this does not affect the edit quality.

5.5 Conclusion

We introduced `DIFFEDIT`, a novel algorithm for semantic image editing based on diffusion models. Given a textual query, using the diffusion model, `DIFFEDIT` infers the relevant regions to be edited rather than requiring a user generated mask. Furthermore, in contrast to other diffusion-based methods, we initialize the generation process with a DDIM encoding of the input image which allows preserving more appearance information from the input image. We provide theoretical analysis that motivates this choice, and show experimentally that this approach conserves more appearance information from the input image, leading to lighter edits. Quantitative and qualitative evaluations on ImageNet, COCO, and images generated by Imagen, show that our approach leads excellent edits, improving over previous approaches. Although `DIFFEDIT` works better with a reference text describing the input image, we believe this additional information can be inferred from input image and target caption. Finally, `DIFFEDIT` demonstrates that there are rich interactions between the text conditioning and spatial arrangement in generated images. In the next chapter, we study in more detail these relationships between semantics and spatial structure.

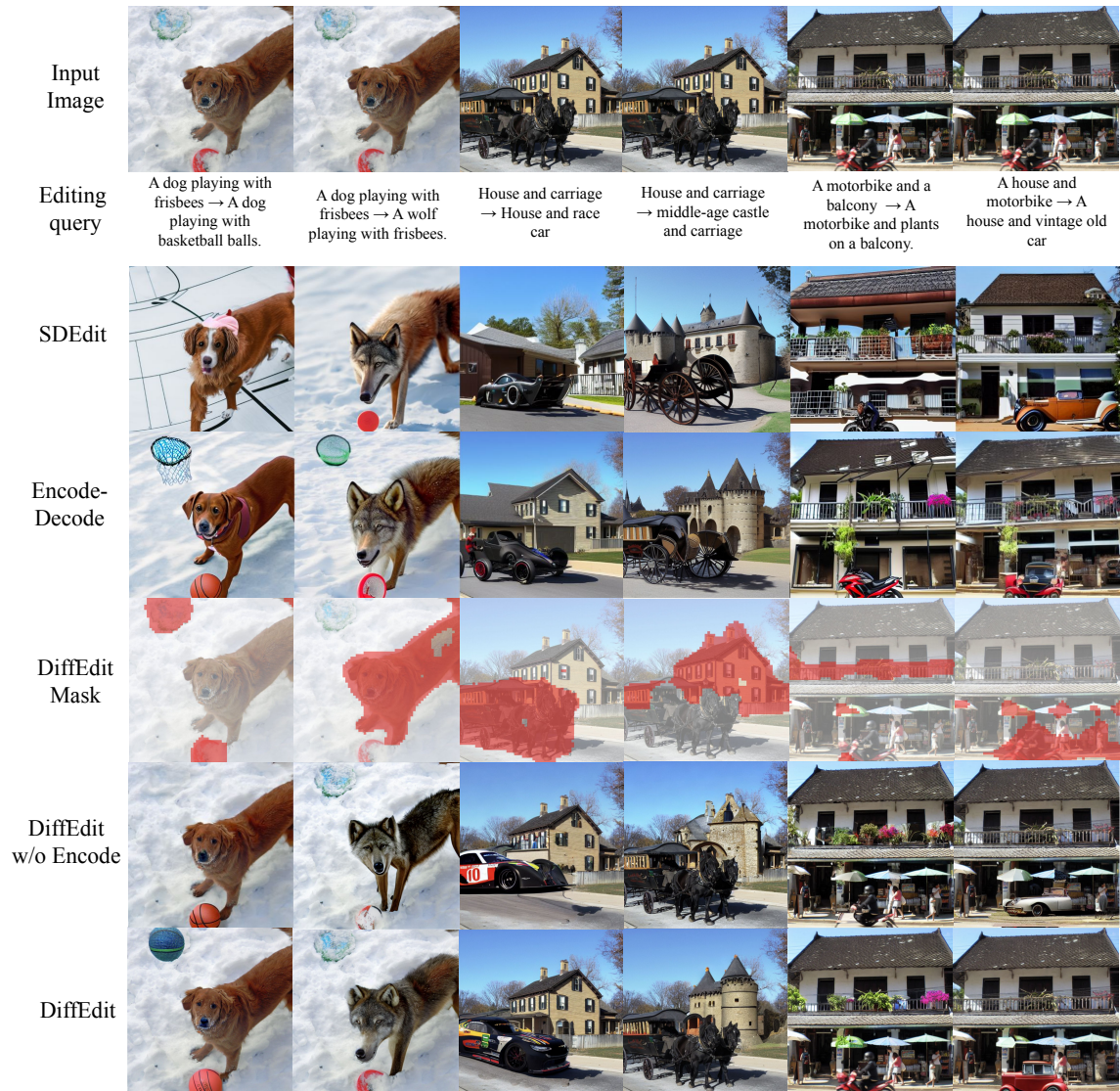


Figure 5.16. – More qualitative examples on COCO. In the first column, the color of the objects to be edited is maintained, which would not be the case with regular inpainting methods. Contrasting similar text query and reference text allows to select the object to be edited.

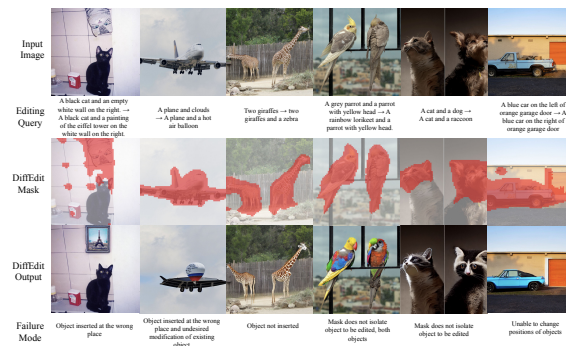


Figure 5.17. – Illustration of failure modes. In the first two columns show difficulty to insert an object in a smooth region of the image. In column three the mask fails to identify a region where to add the zebra. Columns 4 and 5 show mask identification errors, where multiple similar objects are included in the mask, whereas matching the text query only requires to edit a single object. In both cases this results in over-editing. Col. 6 shows the failure to change a spatial relation in the image.

IMAGE SYNTHESIS FROM SEMANTIC SEGMENTATION MAPS

6.1 Introduction

In the previous chapters, we have studied how textual editing prompts can be used to condition image generation. Text prompts can effectively convey information about the objects in the scene, their interactions, and the overall style of the image; however they may not be the optimal choice for achieving fine-grained spatial control. Accurately describing the pose, position, and shape of each object in a complex scene with words can be a cumbersome task. Moreover, recent works have shown the limitation of diffusion models to follow spatial guidance expressed in natural language (Avrahami et al. 2022a; Paga et al. 2022).

On the other hand, semantic image synthesis is a conditional image generation task that allows for detailed spatial control, by providing a semantic map to indicate the desired class label for each pixel. Both adversarial (T. Park et al. 2019; Schönfeld et al. 2021) and diffusion-based (T. Wang et al. 2022a; W. Wang et al. 2022) approaches have been explored to generate high-quality and diverse images. However, these approaches rely heavily on large datasets with tens to hundreds of thousands of images annotated with pixel-precise label maps, which are expensive to acquire and inherently limited in the number of class labels.

In this chapter, we propose a zero-shot approach semantic image synthesis called **ZESTGUIDE**, short for **Z**ErO-shot **S**egmenTation **G**UIDance, which empowers a pretrained text-to-image diffusion model to enable image generation conditioned on segmentation maps with corresponding free-form textual descriptions. A few examples are shown in Figure 6.1. The task of semantic image synthesis is not an image editing task but rather a conditional generation task. It is nonetheless very close to approaches presented in previous chapters for two reasons: first, being able to synthesize an object at a precise location is very useful in the context of image editing and can be combined with other image editing tools such as inpainting; second, our goal is in spirit similar to Chapter 5, since we augment diffusion models with novel abilities at test time, in a zero-shot fashion.



Figure 6.1. – ZestGuide generates images conditioned on segmentation maps with corresponding free-form textual descriptions.

Our zero-shot approach builds upon classifier-guidance techniques that allow to adapt pretrained diffusion models (Dhariwal and Nichol 2021a) for conditional generation, as presented in Chapter 2. These techniques utilize an external classifier to steer the iterative denoising process of diffusion models toward the generation of an image corresponding to the condition. While these approaches have been successfully applied to various forms of conditioning, such as class labels (Dhariwal and Nichol 2021a) and semantic maps (Bansal et al. 2023), they still rely on pretrained recognition models. In the case of semantic image synthesis, this means that an image-segmentation network must be trained, which (i) violates our zero-shot objective, and (ii) allows each segment only to be conditioned on a single class label. To circumvent the need for an external classifier, our approach takes advantage of the spatial information embedded in the cross-attention layers of the diffusion model to achieve zero-shot image segmentation. Guidance is then achieved by comparing a segmentation extracted from the attention layers with the conditioning map, eliminating the need for an external segmentation network. In particular, ZESTGUIDE computes a loss between the inferred segmentation and the input segmentation, and uses the gradient of this loss to guide the noise estimation process, allowing conditioning on free-form text rather than just class labels. No fine-tuning of the text-to-image diffusion model is required. See Figure 6.2 for an overview of ZESTGUIDE.

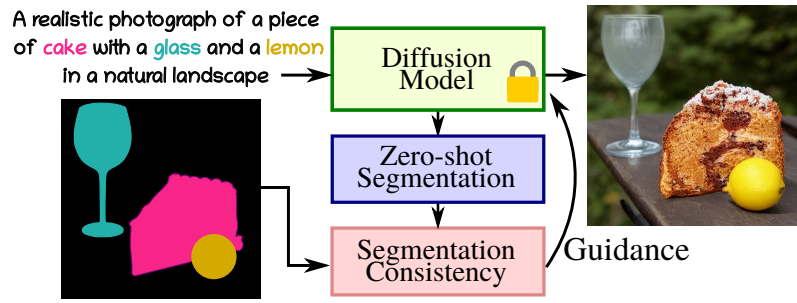


Figure 6.2. – In ZESTGUIDE, the image generation is guided by the gradient of a loss computed between the input segmentation and a segmentation recovered from attention in a text-to-image diffusion model.

The remainder of the chapter is organized as follows: after going over related work, we present our Zestguide algorithm, and notably how the attention maps of the cross-attention layer are used to perform zero-shot segmentation. We then perform qualitative experiments on COCO, improving over existing both zero-shot and training-based approaches both quantitatively and qualitatively.

6.2 Related work

In this section, we go over the literature of generative models conditioned on spatial semantic maps, as well as related work on train-free adaptation of diffusion models.

Spatially conditioned generative image models Following seminal works on image-to-image translation (Isola et al. 2017), spatially constrained image generation has been extensively studied. In particular, the task of semantic image synthesis consists in generating images conditioned on masks where each pixel is annotated with a class label. Until recently, GAN-based approaches were prominent with methods such as SPADE (T. Park et al. 2019), and OASIS (Schönfeld et al. 2021). Alternatively, autoregressive transformer models over discrete VQ-VAE (A. van den Oord et al. 2017) representations to synthesize images from text and semantic segmentation maps have been considered (Esser et al. 2021a; Gafni et al. 2022b; Razavi et al. 2019), as well as non-autoregressive models with faster sampling (Chang et al. 2022; Lezama et al. 2022).

Diffusion models have also been explored for semantic image synthesis. For example, PITI (T. Wang et al. 2022a) finetunes GLIDE (Nichol et al. 2022), a large pretrained text-to-image generative model, by replacing its text encoder with an encoder of semantic segmentation maps. SDM (W. Wang et al. 2022) trains a

diffusion model using SPADE blocks to condition the denoising U-Net on the input segmentation.

As introduced in Chapter 2, the generative algorithm of diffusion models can be guided to match a specific classification result given by a pretrained classifier. For semantic image synthesis, the gradient of a pretrained semantic segmentation network can be used as guidance (Bansal et al. 2023). This approach, however, suffers from two drawbacks. First, only the classes recognized by the segmentation model can be used to constrain the image generation, although this can to some extent be alleviated using an open-vocabulary segmentation model like CLIPSeg (Lüddecke and Ecker 2022). The second drawback is that this approach requires a full forwards-backwards pass through the external segmentation network in order to obtain the gradient at each step of the diffusion process, which requires additional memory and compute on top of the diffusion model itself.

While there is a vast literature on semantic image synthesis, it is more limited when it comes to the more general task of synthesizing images conditioned on masks with free-form textual descriptions. SpaText (Avrahami et al. 2022a) fine-tunes a large pretrained text-to-image diffusion model with an additional input of segments annotated with free-form texts. This representation is extracted from a pretrained multi-modal CLIP encoder (Radford et al. 2021b): using visual embeddings during training, and swapping to textual embeddings during inference. GLIGEN (Yuheng Li et al. 2023) adds trainable layers on top of a pretrained diffusion models to extend conditioning from text to bounding boxes and pose. These layers take the form of additional attention layers that incorporate the local information. T2I (Mou et al. 2023) and ControlNet (Lvmin Zhang and Agrawala 2023) propose to extend a pretrained and frozen diffusion model with small adapters for task-specific spatial control using pose, sketches, or segmentation maps. All these methods require to be trained on a large dataset with segmentation annotations, which is computationally costly and requires specialized training data.

Train-free adaptation of text-to-image diffusion models Several recent studies (Chefer et al. 2023; W. Feng et al. 2022; Hertz et al. 2022; Parmar et al. 2023) found that the positioning content in generated images from large text-to-image diffusion models correlates with the cross-attention maps, which diffusion models use to condition the denoising process on the conditioning text. This correlation can be leveraged to adapt text-to-image diffusion at inference time for various downstream applications. For example, (Chefer et al. 2023; W. Feng et al. 2022) aim to achieve better image composition and attribute binding. Feng *et al.* (W. Feng et al. 2022) design a pipeline to associate attributes to objects and incorporate this linguistic structure by modifying values in cross-attention maps. Chefer *et al.* (Chefer et al. 2023) guide the generation process with gradients from a loss aim-

ing at strengthening attention maps activations of ignored objects. Closer to our work, eDiff-I (Balaji et al. 2022) proposes a procedure to synthesize images from segmentation maps with local free-form texts. They do so by rescaling attention maps at locations specified by the input semantic masks. MultiDiffusion (Bar-Tal et al. 2023) fuses multiple generation processes constrained by shared parameters from a pretrained diffusion model by solving an optimization problem, and applying it to panorama generation and spatial image guidance. Finally, in (Bansal et al. 2023), a pretrained segmentation net guides image generation to respect a segmentation map during the denoising process of the diffusion model.

6.3 ZestGuide algorithm

In this section, we present ZESTGUIDE, which extends pretrained text-to-image diffusion models to enable conditional generation of images based on segmentation maps and associated text without requiring additional training, as described in Section 6.3.2. In Figure 6.3 we provide an overview of ZESTGUIDE.

6.3.1 Classifier Guidance

Classifier guidance is a technique for conditional sampling of diffusion models (Sohl-Dickstein et al. 2015; Yang Song et al. 2021), which we have presented in Chapter 2, equation 6.1. We recall here the equation for sampling an image that will be associated the label c by an external classifier with probability distribution p_π :

$$\begin{aligned} \tilde{\epsilon}_\theta(\mathbf{x}_t, t, \rho(T)) &= \epsilon_\theta(\mathbf{x}_t, t, \rho(T)) \\ &\quad - \sqrt{1 - \alpha_t} \nabla_{\mathbf{x}_t} p_\pi(c|\mathbf{x}_t), \end{aligned} \tag{6.1}$$

where $\rho(T)$ is an encoding of the conditioning text information provided as input to the diffusion model. Classifier guidance can be straightforwardly adapted to generate images conditioned on semantic segmentation maps by replacing the classifier by a segmentation network which outputs a label distribution for each pixel in the input image. However this approach suffers from several weaknesses: (i) it requires to train an external segmentation model; (ii) semantic synthesis is bounded to the set of classes modeled by the segmentation model; (iii) it is computationally expensive since it implies back-propagation through both the latent space decoder and the segmentation network at every denoising step. To address these issues, we propose to employ the cross-attention maps computed in the denoising model ϵ_θ of text-to-image diffusion models to achieve zero-shot

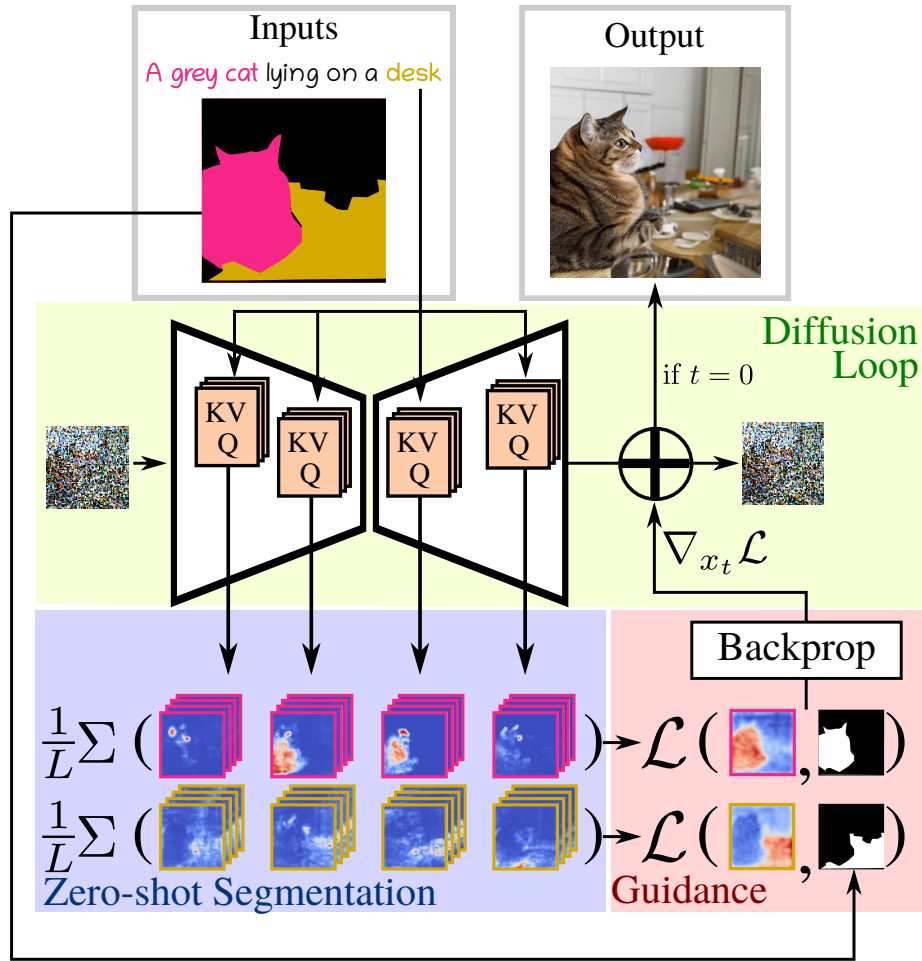


Figure 6.3. – ZESTGUIDE extracts segmentation maps from text-attention layers in pretrained diffusion models, and uses them to align the generation with input masks via gradient-based guidance.

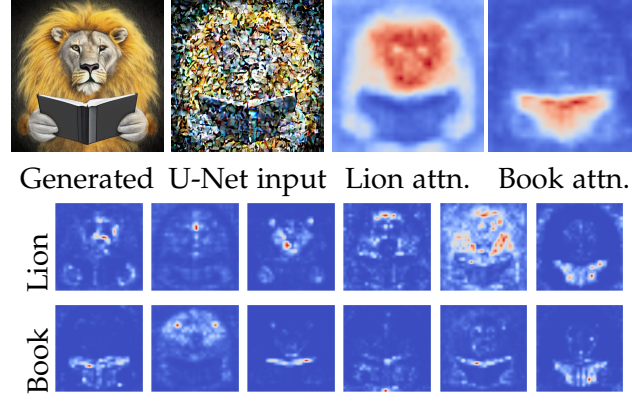


Figure 6.4. – Top, from left to right: image generated from the prompt “A lion reading a book.”, the noisy input to the U-Net at $t = 20$, cross-attention averaged over different heads and U-Net layers for “Lion” and “Book”. Bottom: individual attention heads.

segmentation. This has two major advantages: first, there is no need to decode the image at each denoising step; second, our zero-shot segmentation process is extremely lightweight, so the additional computational cost almost entirely comes from back-propagation through the U-Net, which is a relatively low-cost method for incorporating classifier guidance.

6.3.2 Zero-shot segmentation with attention

To condition the image generation, we consider a text prompt of length N denoted as $T = \{\tau_1, \dots, \tau_N\}$, and a set of K binary segmentation maps $\mathbf{S} = \{\mathbf{S}_1, \dots, \mathbf{S}_K\}$. Each segment \mathbf{S}_i is associated with a subset $T_i \subset T$. The diffusion model has been trained conditionally to the text token embeddings, which are processed with cross-attention layers in the U-Net, where keys and values are computed from the text embeddings.

Attention map extraction We leverage cross-attention layers of the diffusion U-Net to segment the image as it is generated. The attention maps are computed independently for every layer and head in the U-Net. For layer l , the queries \mathbf{Q}_l are computed from local image features using a linear projection layer. Similarly, the keys \mathbf{K}_l are computed from the word descriptors T with another layer-specific linear projection. The cross-attention from image features to text tokens, is computed as

$$\mathbf{A}_l = \text{Softmax} \left(\frac{\mathbf{Q}_l \mathbf{K}_l^T}{\sqrt{d}} \right), \quad (6.2)$$

where the query/key dimension d is used to normalize the softmax energies (A. Vaswani et al. 2017). Let $\mathbf{A}_l^n = \mathbf{A}_l[n]$ denote the attention of image features w.r.t. specific text token $T_n \in T$ in layer l of the U-Net. To simplify notation, we use l to index over both the layers of the U-Net and the different attention heads in each layer. In practice, we find that the attention maps provide meaningful localization information, but only when they are averaged across different attention heads and feature layers. See Figure 6.4 for an illustration.

Since the attention maps have varying resolutions depending on the layer, we up-sample them to the highest resolution. Then, for each segment we compute an attention map \mathbf{S}_i by averaging attention maps across layers and text tokens associated with the segment:

$$\hat{\mathbf{S}}_i = \frac{1}{L} \sum_{l=1}^L \sum_{j=1}^N \mathbb{I}[\tau_j \in T_i] \mathbf{A}_j^l, \quad (6.3)$$

where $\mathbb{I}[\cdot]$ is the Iverson bracket notation which is one if the argument is true and zero otherwise.

Spatial self-guidance We compare the averaged attention maps to the input segmentation using a sum of binary cross-entropy losses computed separately for each segment:

$$\mathcal{L}_{\text{Zest}} = \sum_{i=1}^K \left(\mathcal{L}_{\text{BCE}}(\hat{\mathbf{S}}_i, \mathbf{S}_i) + \mathcal{L}_{\text{BCE}}\left(\frac{\hat{\mathbf{S}}_i}{\|\hat{\mathbf{S}}_i\|_\infty}, \mathbf{S}_i\right) \right). \quad (6.4)$$

In the second loss term, we normalized the attention maps $\hat{\mathbf{S}}_i$ independently for each object. This choice is motivated by two observations. Firstly, we found that averaging softmax outputs across heads, as described in Equation equation 6.3, generally results in low maximum values in $\hat{\mathbf{S}}_i$. By normalizing the attention maps, we make them more comparable with the conditioning \mathbf{S} . Secondly, we observed that estimated masks can have different maximum values across different segments resulting in varying impacts on the overall loss. Normalization helps to balance the impact of each object. However, relying solely on the normalized term is insufficient, as the normalization process cancels out the gradient corresponding to the maximum values.

We then use DDIM sampling with classifier guidance based on the gradient of this loss. We use Eq. (6.1) to compute the modified noise estimator at each denoising step. Interestingly, since \mathbf{x}_{t-1} is computed from $\tilde{\epsilon}_\theta(\mathbf{x}_t)$, this conditional DDIM sampling corresponds to an alternation of regular DDIM updates and gradient descent updates on \mathbf{x}_t of the loss \mathcal{L} , with a fixed learning rate η multiplied



Figure 6.5. – ZESTGUIDE generations on coarse hand-drawn masks.

by a function $\lambda(t)$ monotonically decreasing from one to zero throughout the generative process. In this formulation, the gradient descent update writes:

$$\tilde{\mathbf{x}}_{t-1} = \mathbf{x}_{t-1} - \eta \cdot \lambda(t) \frac{\nabla_{\mathbf{x}_t} \mathcal{L}_{\text{Zest}}}{\|\nabla_{\mathbf{x}_t} \mathcal{L}_{\text{Zest}}\|_{\infty}}. \quad (6.5)$$

Note that differently from Eq. (6.1), the gradient is normalized to make updates more uniform in strength across images and denoising steps. We note that the learning rate η can be set freely, which, as noted by (Dhariwal and Nichol 2021a), corresponds to using a renormalized classifier distribution in classifier guidance. As in (Balaji et al. 2022), we define a hyperparameter τ as the fraction of steps during which classifier guidance is applied. Preliminary experiments suggested that classifier guidance is only useful in the first 50% of DDIM steps, and we set $\tau = 0.5$ as our default value, see Section 6.4.3 for more details.

6.4 Experiments

We first present a few examples on hand-drawn masks. Then we present our experimental setup in Section 6.4.1, followed by our main results in Section 6.4.2 and ablations in Section 6.4.3.

Visualizations on hand-drawn masks In Figure 6.5, we show generations conditioned on coarse hand-drawn masks, a setting close to real-world applications. In this case the generated objects do not exactly match the shape of conditioning masks: the flexibility of ZESTGUIDE helps to generate realistic images even in the case of unrealistic segmentation masks, see *e.g.* the cow and mouse examples.

Method	Zero-shot	Eval-all			Eval-filtered			Eval-few		
		↓FID	↑mIoU	↑CLIP	↓FID	↑mIoU	↑CLIP	↓FID	↑mIoU	↑CLIP
OASIS (Schönfeld et al. 2021)	✗	15.0	52.1	—	18.2	53.7	—	46.8	41.4	—
SDM (W. Wang et al. 2022)	✗	17.2	49.3	—	28.6	41.7	—	65.3	29.3	—
SD w/ T2I-Adapter (Mou et al. 2023)	✗	17.2	33.3	31.5	17.8	35.1	31.3	19.2	31.6	30.6
LDM w/ External Classifier	✗	24.1	14.2	30.6	23.2	17.1	30.2	23.7	20.5	30.1
SD w/ SpaText (Avrahami et al. 2022a)	✗	19.8	16.8	30.0	18.9	19.2	30.1	16.2	23.8	30.2
SD w/ PwW (Balaji et al. 2022)	✓	36.2	21.2	29.4	35.0	23.5	29.5	25.8	23.8	29.6
LDM w/ MultiDiff.(Bar-Tal et al. 2023)	✓	59.9	15.8	23.9	46.7	18.6	25.8	21.1	19.6	29.0
LDM w/ PwW	✓	22.9	27.9	31.5	23.4	31.8	31.4	20.3	36.3	31.2
LDM w/ ZESTGUIDE (ours)	✓	22.8	33.1	31.9	23.1	43.3	31.3	21.0	46.9	30.3

Table 6.1. – Comparison of ZESTGUIDE to other methods in our three evaluation settings. OASIS and SDM are trained from scratch on COCO, other methods are based on pretrained text-to-image models: StableDiffusion (SD) or our latent diffusion model (LDM). Methods that do not allow for free-form text description of segments are listed in the upper part of the table. Best scores in each part of the table are marked in bold. For OASIS and SDM the CLIP score is omitted as it is not meaningful for methods that don’t condition on text prompts.

6.4.1 Experimental setup

Evaluation protocol We use the COCO-Stuff validation split, which contains 5k images annotated with fine-grained pixel-level segmentation masks across 171 classes, and five captions describing each image (Caesar et al. 2018). We adopt three different setups to evaluate our approach and to compare to baselines. In all three settings, the generative diffusion model is conditioned on one of the five captions corresponding to the segmentation map, but they differ in the segmentation maps used for spatial conditioning.

The first evaluation setting, *Eval-all*, conditions image generation on complete segmentation maps across all classes, similar to the evaluation setup in OASIS (Schönfeld et al. 2021) and SDM (W. Wang et al. 2022). In the *Eval-filtered* setting, segmentation maps are modified by removing all segments occupying less than 5% of the image, which is more representative of real-world scenarios where users may not provide segmentation masks for very small objects. Finally, in *Eval-few* we retain between one and three segments, each covering at least 5% of the image, similar to the setups in (Avrahami et al. 2022a; Bar-Tal et al. 2023). It is the most realistic setting, as users may be interested in drawing only a few objects, and therefore the focus of our evaluation. Regarding the construction of the text prompts, we follow (Avrahami et al. 2022a) and concatenate the annotated prompt of COCO with the list of class names corresponding to the input segments.

Evaluation metrics We use the two standard metrics to evaluate semantic image synthesis, see *e.g.* (T. Park et al. 2019; Schönfeld et al. 2021). Fréchet Inception Distance (FID) (Heusel et al. 2017b) captures both image quality and diversity. We compute FID with InceptionV3 and generate 5k images. The reference set is the original COCO validation set, and we use code from (Parmar et al. 2022). The mean Intersection over Union (mIoU) metric measures to what extent the generated images respect the spatial conditioning. We compute the mIoU metric with ViT-Adapter (Z. Chen et al. 2023) as segmentation model rather than the commonly used DeepLabV2 (L.-C. Chen et al. 2015), as the former improves over the latter by 18.6 points of mIoU (from 35.6 to 54.2) on COCO-Stuff. We additionally compute a CLIP score that measures alignment between captions and generated images. All methods, including ours, generate images at resolution 512×512 , except OASIS and SDM, for which we use available pretrained checkpoints synthesizing images at resolution 256×256 , which we up-sample to 512×512 .

Baselines We compare to baselines that are either trained from scratch, fine-tuned or training-free. The adversarial OASIS model (Schönfeld et al. 2021) and diffusion-based SDM model (W. Wang et al. 2022) are both trained from scratch and conditioned on segmentation maps with classes of COCO-Stuff dataset. For SDM we use $T = 50$ diffusion decoding steps. T2I-Adapter (Mou et al. 2023) and SpaText (Avrahami et al. 2022a) both fine-tune pre-trained text-to-image diffusion models for spatially-conditioned image generation by incorporating additional trainable layers in the diffusion pipeline. Similar to Universal Guidance (Bansal et al. 2023), we implemented a method in which we use classifier guidance based on the external pretrained segmentation network DeepLabV2 (Liang-Chieh Chen et al. 2017) to guide the generation process to respect a semantic map. We also compare ZESTGUIDE to other zero-shot methods that adapt a pre-trained text-to-image diffusion model during inference. MultiDiffusion (Bar-Tal et al. 2023) decomposes the denoising procedure into several diffusion processes, where each one focuses on one segment of the image and fuses all these different predictions at each denoising iteration. In (Balaji et al. 2022) a conditioning pipeline called “*paint-with-words*” (PwW) is proposed, which manually modifies the values of attention maps. For a fair comparison, we evaluate these zero-shot methods on the same diffusion model used to implement our method. Note that SpaText, MultiDiffusion, PwW, and our method can be locally conditioned on free-form text, unlike Universal Guidance, OASIS, SDM and T2I-Adapter which can only condition on COCO-Stuff classes.

Text-to-image model Due to concerns regarding the training data of Stable Diffusion (such as copyright infringements and consent), we refrain from experi-

menting with this model and instead use a large diffusion model (2.2B parameters) trained on a proprietary dataset of 330M image-text pairs. We refer to this model as LDM. Similar to (Rombach et al. 2022a) the model is trained on the latent space of an autoencoder, and we use an architecture for the diffusion model based on GLIDE (Nichol et al. 2022), with a T5 text encoder (Raffel et al. 2022). With an FID score of 19.1 on the COCO-stuff dataset, our LDM model achieves image quality similar to that of Stable Diffusion (Rombach et al. 2022a), whose FID score was 19.0, while using an order of magnitude less training data.

Implementation details For all experiments that use our LDM diffusion model, we use 50 steps of DDIM sampling with classifier-free guidance strength set to 3. For ZESTGUIDE results, unless otherwise specified, we use classifier guidance in combination with the PwW algorithm. We review this design choice in Section 6.4.3.

6.4.2 Main results

We present our evaluation results in Table 6.1. Compared to other methods that allow free-text annotation of segments (bottom part of the table), our approach leads to marked improvements in mIoU in all settings. For example improving by more than 10 points (36.3 to 46.9) over the closest competitor PwW, in the most realistic Eval-few setting. Note that we even improve over SpaText, which finetunes Stable Diffusion specifically for this task. In terms of CLIP score, our approach yields similar or better results across all settings. Our approach obtains the best FID values among the methods based on our LDM text-to-image model. SpaText obtains the best overall FID values, which we attribute to the fact that it is finetuned on a dataset very similar to COCO, unlike the vanilla Stable Diffusion or our LDM.

In the top part of the table we report results for methods that do not allow to condition segments on free-form text, and all require training on images with semantic segmentation maps. We find they perform well in the Eval-all setting for which they are trained, and also in the similar Eval-filtered setting, but deteriorate in the Eval-few setting where only a few segments are provided as input. In the Eval-few setting, our ZESTGUIDE approach surpasses all methods in the top part of the table in terms of mIoU. Compared to LDM w/ External Classifier, which is based on the same diffusion model as ZESTGUIDE but does not allow conditioning segments on free text, we improve across all metrics and settings, while being much faster at inference: LDM w/ ExternalClassifier takes 1 min. for one image while ZESTGUIDE takes around 15 secs.



Figure 6.6. – Qualitative comparison of ZestGuide to other methods based on LDM, conditioning on COCO captions and up to three segments.

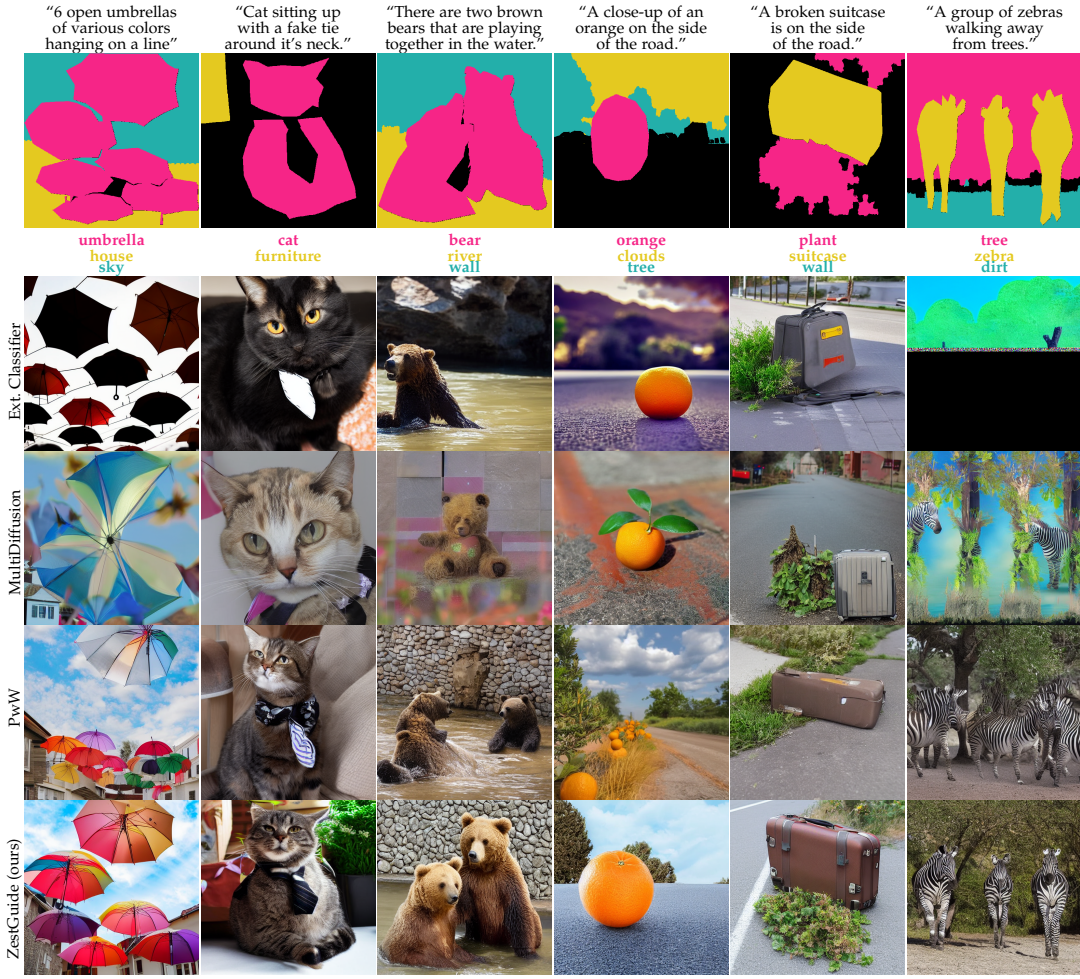


Figure 6.7. – Qualitative comparison of ZestGuide to other methods based on LDM, conditioning on COCO captions and up to three segments.

We provide qualitative results for the methods based on LDM in Figure 6.6 when conditioning on up to three segments, corresponding to the Eval-few setting. Our ZESTGUIDE clearly leads to superior alignment between the conditioning masks and the generated content.

6.4.3 Ablations

We first present a visualization of the impact of Zestguide across time steps, before moving to quantitative ablations on COCO.

Evolution of attention maps across time steps We show in Figure 6.8 average attention maps on the different objects present in the input segmentation during the first 12 denoising steps with and without our guidance scheme. We condition

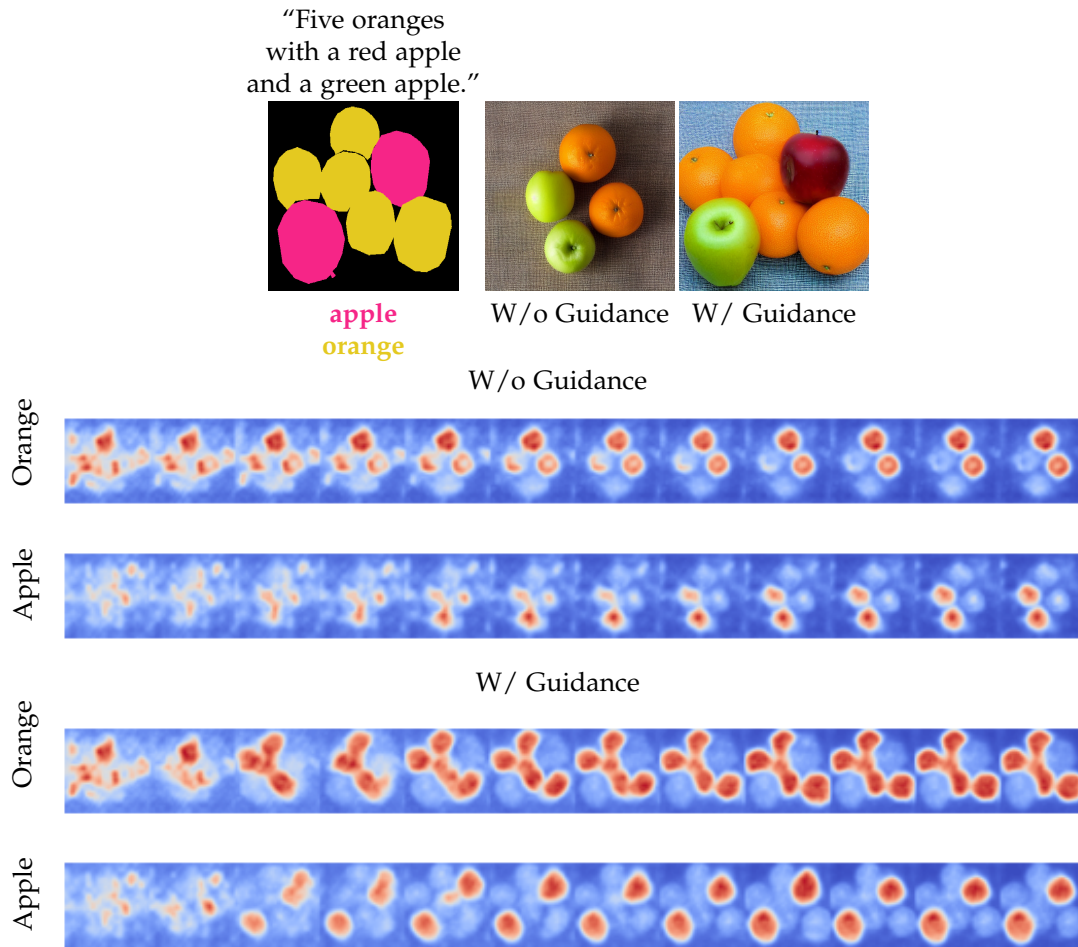


Figure 6.8. – Visualization of first 12 denoising steps out of 50 steps. Same seed for w/ and w/o guidance.

on the same Gaussian noise seed in both cases. We notice that attention maps quickly converges to the correct input conditioning mask when we apply ZEST-GUIDE and that the attention masks are already close to ground truth masks only after 12 denoising iteration steps out of 50.

Next, we perform quantitative ablations, focusing on evaluation settings *Eval-filtered* and *Eval-few*, which better reflect practical use cases. To reduce compute, metrics are computed with a subset of 2k images from the COCO val set.

Ablation on hyperparameters τ and η Our approach has two hyperparameters that control the strength of the spatial guidance: the learning rate η and the percentage of denoising steps τ until which classifier guidance is applied. Varying these hyperparameters strikes different trade-offs between mIoU (better with stronger guidance) and FID (better with less guidance and thus less perturbation

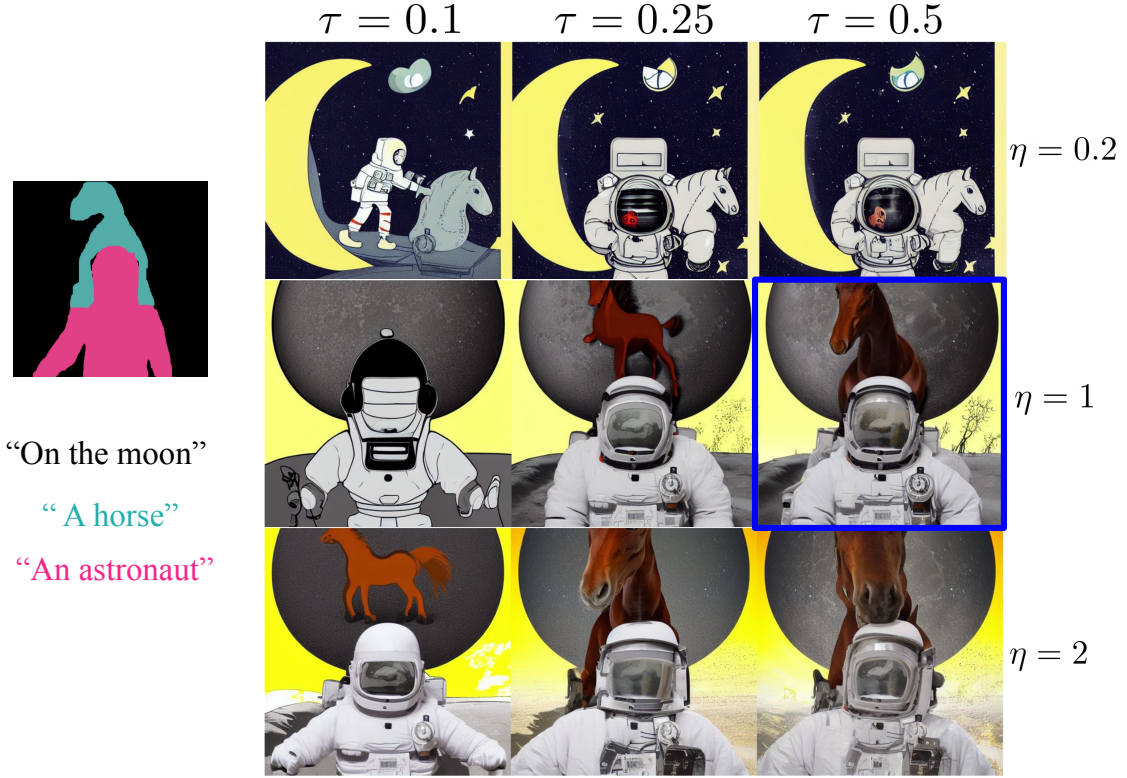


Figure 6.9. – ZESTGUIDE outputs when varying the two main hyperparameters η (learning rate) and τ (percentage of steps using classifier guidance). Our default configuration is $\eta = 1, \tau = 0.5$.

of the diffusion model). In Figure 6.9 we show generations for a few values of these parameters. We can see that, given the right learning rate, applying gradient updates for as few as the first 25% denoising steps can suffice to enforce the layout conditioning. This is confirmed by quantitative results in the Eval-few setting presented in paragraph 6.3. For $\eta = 1$, setting $\tau = 0.5$ strikes a good trade-off with an mIoU of 43.3 and FID of 31.5. Setting $\tau = 1$ marginally improves mIoU by 1.3 points, while worsening FID by 3.2 points, while setting $\tau = 0.1$ worsens mIoU by 9.1 points for a gain of 1 point in FID. Setting $\tau = 0.5$ requires additional compute for just the first half of denoising steps, making our method in practice only roughly 50% more expensive than regular DDIM sampling.

Guidance losses and synergy with PwW In Figure 6.10 we explore the FID-mIoU trade-off in the Eval-filtered setting, for PwW and variations of our approach using different losses and with/without including PwW. The combined loss refers to our full loss in Eq. (6.4), while the BCE loss ignores the second normalized loss. For PwW, the FID-mIoU trade-off is controlled by the constant W that is added to the attention values to reinforce the association of image regions and

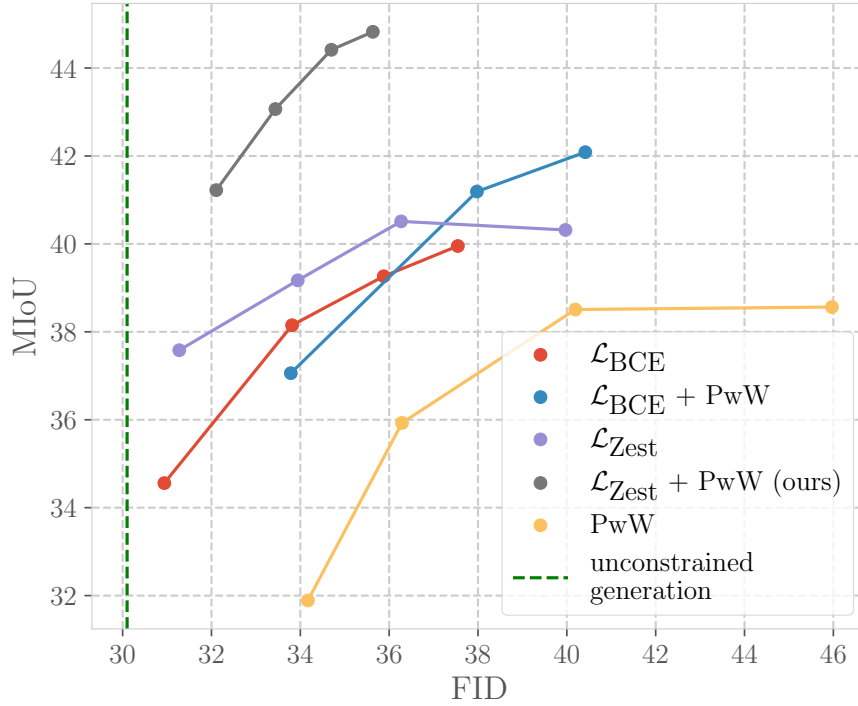


Figure 6.10. – Trade-off in *Eval-filtered* setting between FID (lower is better) and mIoU (higher is better) of PwW and ZESTGUIDE using different losses. In dotted green is shown the FID for unconstrained text-to-image generation. Using \mathcal{L}_{Zest} in combination with PwW (our default setting) gives the best trade-off.

their corresponding text. For ZESTGUIDE, we vary η to obtain different trade-offs, with $\tau = 0.5$. We observe that all versions of our approach provide better mIoU-FID trade-offs than PwW alone. Interestingly, using the combined loss and PwW separately hardly improve the mIoU-FID trade-off w.r.t. only using the BCE loss, but their combination gives a much better trade-off (Combined Loss + pWW). This is possibly due to the loss with normalized maps helping to produce more uniform segmentation masks, which helps PwW to provide more consistent updates.

In the remainder of the ablations, we consider the simplest version of ZESTGUIDE with the \mathcal{L}_{BCE} loss and without PwW, to better isolate the effect of gradient guiding.

Attention map averaging As mentioned in Section 6.3.2, we found that averaging the attention maps across all heads of the different cross-attention layers is important to obtain good spatial localization. We review this choice in Table 6.2. When we compute our loss on each head separately, we can see a big drop in

Components	\downarrow FID	\uparrow mIoU	\uparrow CLIP
Loss for each attention head	33.6	32.1	29.9
Loss for each layer	31.6	42.7	30.5
Loss for global average (ours)	31.5	43.3	30.4

Table 6.2. – Evaluation of ZESTGUIDE on Eval-few setting, with different averaging schemes for computing the loss. Averaging all attention heads before applying the loss gives best results.

mIoU scores (-11 points). This reflects our observation that each attention head focuses on different parts of each object. By computing a loss on the averaged maps, a global pattern is enforced while still maintaining flexibility for each attention head. This effect is much less visible when we average attention maps per layer, and apply the loss per layer: in this case mIoU deteriorates by 1.6 points, while FID improves by 0.9 points.

Gradient normalization Unlike standard classifier guidance, ZESTGUIDE uses normalized gradient to harmonize gradient descent updates in Eq. (6.5). We find that while ZESTGUIDE also works without normalizing gradient, adding it gives a boost of 2 mIoU points for comparable FID scores. Qualitatively, it helped for some cases where the gradient norm was too high at the beginning of generation process, which occasionally resulted in low-quality samples.

Impact of parameter τ In our method, classifier guidance is only used in a fraction τ of denoising steps, after which it is disabled. Table 6.3 demonstrates that after our default value $\tau = 0.5$, mIoU gains are marginal, while the FID scores are worse. Conversely, using only 10% or 25% of denoising steps for classifier guidance already gives very good mIoU/FID scores, better than PwW for $\tau = 0.25$. As illustrated in Figure 6.8, this is because estimated segmentation maps converge very early in the generation process.

Components	\downarrow FID	\uparrow mIoU	\uparrow CLIP
$\tau = 0.1$	30.54	34.25	31.18
$\tau = 0.25$	30.36	40.75	30.77
$\tau = 0.5$	31.53	43.34	30.44
$\tau = 1$	34.75	44.58	29.99

Table 6.3. – Ablation on parameter τ , with fixed learning rate $\eta = 1$ in the Eval-few setting.

Tokens used as attention keys Our estimated segmentation masks are computed with an attention mechanism over a set of keys computed from the text prompt embeddings. In this experiment, we analyze whether the attention over the full text-prompt is necessary, or whether we could simply use classification scores over the set of classes corresponding to the segments. We encode each class text separately with the text encoder, followed by average pooling to get a single embedding per class. Computing our loss with these embeddings as attention keys results in a probability distribution over the segmentation classes. We find that the FID scores are worse (+ 3 pts FID), but the mIoU scores are very close (43.36 vs 43.34). We conclude that our loss function primarily serves to align spatial image features with the relevant textual feature at each spatial location, and that the patterns that we observe in attention maps are a manifestation of this alignment.

Attention layers used We first validate which layers are useful for computing our classifier guidance loss in Table 6.4. We find that whatever the set of cross-attention layers used for computing loss, the mIoU and FID scores are very competitive. In accordance with preliminary observations, it is slightly better to skip attention maps at resolution 8 when computing our loss.

Layers used	↓FID	↑mIoU	↑CLIP
All layers	33.74	40.17	30.19
Only decoder layers	33.81	40.02	30.05
Only encoder layers	30.98	38.24	30.67
Only res32 layers	29.35	39.49	30.75
Only res16 layers	33.59	40.27	30.23
res16 and res32 layers (ours)	31.53	43.34	30.44

Table 6.4. – Ablation on cross-attention layers used for estimating segmentation maps.

6.5 Conclusion

In this chapter, we have presented ZESTGUIDE, a zero-shot method which enables precise spatial control over the generated content by conditioning on segmentation masks annotated with free-form textual descriptions. Our approach leverages implicit segmentation maps extracted from text-attention in pre-trained text-to-image diffusion models to align the generation with input masks. Experimental results demonstrate that our approach achieves high-quality image generation while accurately aligning the generated content with input segmenta-

tions. Our quantitative evaluation shows that ZESTGUIDE is even competitive with methods trained on large image-segmentation datasets.

Despite this success, there remains a limitation shared by many existing approaches. Specifically, the current approach, like others, tends to overlook small objects in the input conditioning maps, which may be related to the low resolution of the attention maps in the diffusion model.

CONCLUSION

We first summarize the contributions that we propose in this thesis before discussing research directions for future work. The main task that we tackle is Text-based Image Editing: we aim at editing images given an instruction written in natural language. We recall here the main challenges: first, it requires joint image/text processing, since the input data is inherently multimodal; second, training data is very scarce and costly to acquire, therefore we need to design zero-shot algorithm that leverage the knowledge embedded in machine learning models pretrained on large-scale image/text datasets.

7.1 Summary of contributions

The first direction we explore in Chapter 3 is a simplified image editing setup which does not require a generative model. We leverage the multimodal embedding space of pretrained image/text contrastive models like CLIP (Radford et al. 2021a). Inspired by the geometric properties of word embedding spaces, we study the suitability of such multimodal embedding spaces for embedding arithmetic. We design a test dataset for retrieval-based image editing, dubbed SIMAT, which provides a quantitative framework for our analysis. Based on this evaluation setup, we show that multimodal embedding trained for image/text retrieval are not the best choice for embedding arithmetic, and that a simple fine-tuning scheme can bring large improvements.

While SIMAT allowed us to study embedding arithmetic, the space of possible images is vastly bigger than any fixed-size dataset, which limits possibilities for editing. Therefore, the second direction we explore in Chapter 4 is real image editing, where we aim at modifying the image given as input instead of retrieving an image from an existing database. Generative Adversarial Networks are able to generate photo-realistic images on narrow domains, but zero-shot editing for real images in the latent space is difficult. To broaden the scope of possible edits, we propose the FLEXIT algorithm: similarly to Chapter 3, we compute a high-level objective of what the edited image should look like. This objective is a multimodal embedding, computed as a linear combination of the embeddings of

the input image and text transformation query. Then, the input image is iteratively optimized, so that the embedding of the edited image gets closer to the "target" multimodal embedding. We show that optimizing the image in the VQGAN auto-encoder latent space (Esser et al. 2021b) allows for natural editing compared to pixel-based optimization, which leads to adversarial unnatural results. Finally, we show how the editing strength can be controlled with regularization optimization terms.

The third direction we explore in Chapter 5 is to leverage diffusion models for text-based image editing. The optimization scheme presented in 4 is quite costly, and is based on a algorithm trained for image/text matching instead of image generation. As a result, images edited with FLEXIT can lack realism when image parts are not correctly identified by CLIP. We turn to generative models and choose to study how to adapt pretrained diffusion models to the task of text-based image editing. We propose the DIFFEDIT algorithm, which automatically produces an editing mask by finding which image regions need editing. We also incorporate reverse DDIM into DIFFEDIT, and give theoretical bounds showing how the distance w.r.t input image can be controlled.

Finally, we explore in Chapter 6 a problem related to image editing: how to constrain the generative process of diffusion models to take into account a set of objects with their precise location in the image. We propose the ZESTGUIDE algorithm, which generates images conditionally to a semantic segmentation map without any training, and allows for open-vocabulary conditional generation. We demonstrate how to best take advantage of the spatio-semantic patterns in the cross-attention maps of the diffusion model.

7.2 Perspective for future work

We believe that the last two chapters represent promising path towards truly flexible text-based image editing. We emphasize here some limitations of these works, on which further work may bring improvements. The limitations of DIFFEDIT are discussed in Section 5.4.6. Notably, the masks identifying the edit location are still not as precise as they could be: the method could be coupled with segmentation approaches or image priors to increase mask accuracy. Also, the method works best when given a pair of sentences, one describing the input image, and one describing the novel image that we want to get. The two sentences should be very similar, so the user has to provide redundant information. It is possible to solve this problem by asking the user to provide only the textual description of the novel image, and to infer the input image's caption that is closest to this

description. This would provide an appreciable improvement over the original algorithm.

A common limitation of all works presented in this thesis is the difficulty to change the position of objects, which is due to how images are stored and generated on a fixed pixel grid. As a result, two images representing the exact same objects at different spatial locations usually have a high edit distance. A promising approach to solve this problem is to use image generation methods that are based on structured representations like BlobGAN (Epstein et al. 2022), where the position of objects in images are encoded as scalar variables which can be changed easily.

Finally, we believe that there is a lot of potential in augmenting diffusion models to do novel tasks at inference time, like we did in Chapter 5 for image editing and in Chapter 6 for semantic image synthesis. Diffusion models could also be adapted for classification (A. C. Li et al. 2023), semantic segmentation, and video editing (Qi et al. 2023). Their adaptability partly comes from their generic training objective (image denoising), which acts similarly to a self-supervised learning task, allowing the model to learn powerful image representations. Future work on zero-shot adaptation of diffusion models are promising research directions to advance the field of computer vision.

7.3 Societal Impact and Ethical challenges

Image editing with diffusion models trained on web-scraped data like LAION raises several ethical challenges that we wish to discuss here. In particular, it was shown that LAION contains inappropriate content (violence, hate, pornography), along with racist and sexist stereotypes. Furthermore, it was found that diffusion models trained on LAION, such as Imagen, can exhibit social and cultural bias. Therefore, the use of such models can raise ethical concerns, whether the text prompt is intentionally harmful or not. Because image editing is usually performed on real images, there are additional ethical challenges, such as potential skin tone change when editing a person or reinforcing harmful social stereotypes. We believe that open-sourcing editing algorithms in a research context contributes to a better understanding of such problems, and can help the community in efforts to mitigate them in the future. Furthermore, image editing tools could be used with harmful intent such as harassment or propagating fake news. This use, known as deep fakes, has been largely discussed in previous work (Etienne 2021). To mitigate potential misuse, the Stable Diffusion model is released under a license focused on ethical and legal use, stating explicitly that users “must not distribute harmful, offensive, dehumanizing content or otherwise harmful rep-

resentations of people or their environments, cultures, religions, etc. produced with the model weights”. Furthermore, the question of how to align AI models to desired behaviors and to prevent them from going out of bounds is an active area of research, for large language models as well as diffusion models (Ouyang et al. 2022; K. Lee et al. 2023; H. Dong et al. 2023).

Our editing benchmark based on the COCO dataset also has some limitations. COCO has a predominant western cultural bias, and we are therefore evaluating transformations on a small subset of images mostly associated with western culture. Finding relevant transformation prompts for an image is challenging: while we found it relevant to leverage existing annotations based on COCO, we believe that evaluating image editing models on a less culturally biased dataset is needed.

Finally, we remind that the recent trend to scale image generation models training to billions of text-image pairs consumes vital resources, from the materials required to build the graphic cards and data centers, to electricity required to run inference with the generative models. These usages cause CO_2 emissions. Even if the electricity is produced from renewable energies, or compensated from renewable electricity certificates, it can increase the CO_2 intensity of higher-priority electricity usage like heating. In this thesis, we have tried to foster research in lightweight adaptation of diffusion models, which requires much less energy compared to training. However, diffusion models are still inefficient models to run compared to GANs, which requires further research to reduce their carbon footprint, along with government incentives to reduce the total computational power used to train and run generative models.

BIBLIOGRAPHY

- Abdal, Rameen, Peihao Zhu, Niloy Mitra, and Peter Wonka (2021). “Styleflow: Attribute-conditioned exploration of StyleGAN-generated images using conditional continuous normalizing flows”. In: *ACM Trans. Graph.* (cit. on p. 43).
- Alami Mejjati, Youssef, Christian Richardt, James Tompkin, Darren Cosker, and Kwang In Kim (2018). “Unsupervised attention-guided image-to-image translation”. In: *Advances in neural information processing systems* 31 (cit. on p. 18).
- Alayrac, Jean-Baptiste, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. (2022). “Flemingo: a visual language model for few-shot learning”. In: *Advances in Neural Information Processing Systems* 35, pp. 23716–23736 (cit. on p. 2).
- Anwaar, Muhammad Umer, Egor Labintcev, and Martin Kleinstenber (2021). “Compositional Learning of Image-Text Query for Image Retrieval”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1140–1149 (cit. on p. 23).
- Artetxe, Mikel and Holger Schwenk (2019). “Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond”. In: *Transactions of the Association for Computational Linguistics* 7, pp. 597–610 (cit. on pp. 24, 29).
- Avrahami, Omri, Thomas Hayes, Oran Gafni, Sonal Gupta, Yaniv Taigman, Devi Parikh, Dani Lischinski, Ohad Fried, and Xi Yin (2022a). “SpaText: Spatio-Textual Representation for Controllable Image Generation”. In: *arXiv preprint arXiv:2211.14305* (cit. on pp. 89, 92, 98, 99).
- Avrahami, Omri, Dani Lischinski, and Ohad Fried (2022b). “Blended diffusion for text-driven editing of natural images”. In: *CVPR* (cit. on pp. 65, 67).
- Awoyemi, John O, Adebayo O Adetunmbi, and Samuel A Oluwadare (2017). “Credit card fraud detection using machine learning techniques: A comparative analysis”. In: *2017 international conference on computing networking and informatics (ICCNi)*. IEEE, pp. 1–9 (cit. on p. 1).
- Balaji, Yogesh, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu (2022). “eDiff-I: Text-to-Image Diffusion Models with Ensemble of Expert Denoisers”. In: *arXiv preprint arXiv:2211.01324*. URL: <https://arxiv.org/abs/2206.00364> (cit. on pp. 93, 97–99).
- Bansal, Arpit, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein (2023). “Universal Guidance for

- Diffusion Models". In: *arXiv preprint* arXiv:2302.07121 (cit. on pp. 90, 92, 93, 99).
- Bar-Tal, Omer, Lior Yariv, Yaron Lipman, and Tali Dekel (2023). "MultiDiffusion: Fusing Diffusion Paths for Controlled Image Generation". In: *arXiv preprint* arXiv:2302.08113 (cit. on pp. 93, 98, 99).
- Barraco, Manuele, Marcella Cornia, Silvia Cascianelli, Lorenzo Baraldi, and Rita Cucchiara (2022). "The unreasonable effectiveness of CLIP features for image captioning: an experimental analysis". In: *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4662–4670 (cit. on p. 11).
- Batmaz, Zeynep, Ali Yurekli, Alper Bilge, and Cihan Kaleli (2019). "A review on deep learning for recommender systems: challenges and remedies". In: *Artificial Intelligence Review* 52, pp. 1–37 (cit. on p. 1).
- Bau, David, Alex Andonian, Audrey Cui, YeonHwan Park, Ali Jahanian, Aude Oliva, and Antonio Torralba (2021). "Paint by word". In: *arXiv preprint* arXiv:2103.10951 (cit. on p. 67).
- Benny, Yaniv, Tomer Galanti, Sagie Benaim, and Lior Wolf (2021). "Evaluation Metrics for Conditional Image Generation". In: *IJCV* 129, pp. 1712–1731. URL: <https://arxiv.org/abs/2004.12361> (cit. on p. 49).
- Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov (2017). "Enriching word vectors with subword information". In: *Transactions of the Association for Computational Linguistics* 5, pp. 135–146 (cit. on p. 29).
- Brock, Andrew, Jeff Donahue, and Karen Simonyan (2019). "Large Scale GAN Training for High Fidelity Natural Image Synthesis". In: *ICLR*. URL: <https://arxiv.org/abs/1809.11096> (cit. on p. 59).
- Brown, Andrew, Cheng-Yang Fu, Omkar Parkhi, Tamara L Berg, and Andrea Vedaldi (2022). "End-to-End Visual Editing with a Generatively Pre-Trained Artist". In: *arXiv preprint* arXiv:2205.01668 (cit. on p. 4).
- Bubeck, Sébastien, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. (2023). "Sparks of artificial general intelligence: Early experiments with gpt-4". In: *arXiv preprint* arXiv:2303.12712 (cit. on p. 2).
- Caesar, Holger, Jasper Uijlings, and Vittorio Ferrari (2018). "COCO-Stuff: Thing and Stuff Classes in Context". In: *CVPR* (cit. on p. 98).
- Caron, Mathilde, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin (2020). "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments". In: *NeurIPS*. URL: <https://arxiv.org/abs/2006.09882> (cit. on p. 59).
- Carvalho, Thyago P, Fabrizzio AAMN Soares, Roberto Vita, Roberto da P Francisco, João P Basto, and Symone GS Alcalá (2019). "A systematic literature review of machine learning methods applied to predictive maintenance". In: *Computers & Industrial Engineering* 137, p. 106024 (cit. on p. 1).

- Casanova, Arantxa, Marlène Careil, Jakob Verbeek, Michal Drozdal, and Adriana Romero-Soriano (2021). "Instance-Conditioned GAN". In: *NeurIPS*. URL: <https://arxiv.org/abs/2109.05070> (cit. on pp. 56, 59).
- Chang, Huiwen, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman (2022). "MaskGIT: Masked Generative Image Transformer". In: *CVPR*. URL: <https://arxiv.org/abs/2202.04200> (cit. on p. 91).
- Chefer, Hila, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or (2023). "Attend-and-Excite: Attention-Based Semantic Guidance for Text-to-Image Diffusion Models". In: *arXiv preprint arXiv:2301.13826* (cit. on p. 92).
- Chen, L.-C., G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille (2015). "Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs". In: *ICLR* (cit. on p. 99).
- Chen, Liang-Chieh, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille (2017). "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs". In: *PAMI* 40.4, pp. 834–848 (cit. on p. 99).
- Chen, Mark, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever (2020). "Generative pretraining from pixels". In: *International conference on machine learning*. PMLR, pp. 1691–1703 (cit. on p. 13).
- Chen, Zhe, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao (2023). "Vision Transformer Adapter for Dense Predictions". In: *ICLR* (cit. on p. 99).
- Choi, Jooyoung, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon (2021). "ILVR: Conditioning method for denoising diffusion probabilistic models". In: *ICCV* (cit. on pp. 65, 67, 73).
- Choi, Yunjey, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha (2020). "StarGAN v2: Diverse Image Synthesis for Multiple Domains". In: *CVPR* (cit. on p. 43).
- Chong, Min Jin and David Forsyth (2020). "Effectively unbiased fid and inception score and where to find them". In: *CVPR* (cit. on p. 49).
- Collins, Edo, Raja Bala, Bob Price, and Sabine Susstrunk (2020). "Editing in style: Uncovering the local semantics of GANs". In: *CVPR* (cit. on p. 18).
- Couairon, Guillaume, Marlene Careil, Matthieu Cord, Stéphane Lathuillière, and Jakob Verbeek (2023a). "ZestGuide". In: *Under Review* (cit. on pp. 7, 22, 131).
- Couairon, Guillaume, Matthijs Douze, Matthieu Cord, and Holger Schwenk (2022a). "Embedding Arithmetic of Multimodal Queries for Image Retrieval". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, O-DRUM workshop*, pp. 4950–4958 (cit. on pp. 6, 21, 131).
- Couairon, Guillaume, Asya Grechka, Jakob Verbeek, Holger Schwenk, and Matthieu Cord (2022b). "Flexit: Towards flexible semantic image translation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18270–18279 (cit. on pp. 6, 21, 72–74, 131).

- Couairon, Guillaume, Jakob Verbeek, Holger Schwenk, and Matthieu Cord (2023b). "Diffedit: Diffusion-based semantic image editing with mask guidance". In: *International Conference in Learning Representations* (cit. on pp. 7, 22, 131).
- Crowson, Katherine (2021). "CLIP Guided Diffusion HQ 512x512". In: URL: <https://colab.research.google.com/drive/1V66mUeJbXrTuQITvJunvnWVn96FEbSI3> (cit. on pp. 65, 67).
- Crowson, Katherine, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff (2022). "Vqgan-clip: Open domain image generation and editing with natural language guidance". In: *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVII*. Springer, pp. 88–105 (cit. on pp. 3, 15, 65).
- Degrave, Jonas, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, et al. (2022). "Magnetic control of tokamak plasmas through deep reinforcement learning". In: *Nature* 602.7897, pp. 414–419 (cit. on p. 1).
- Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009a). "ImageNet: A Large-Scale Hierarchical Image Database". In: *CVPR09* (cit. on p. 2).
- Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009b). "ImageNet: A Large-Scale Hierarchical Image Database". In: *CVPR* (cit. on p. 48).
- Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, pp. 248–255 (cit. on pp. 9, 72).
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (cit. on p. 10).
- Dhamo, Helisa, Azade Farshad, Iro Laina, Nassir Navab, Gregory D Hager, Federico Tombari, and Christian Rupprecht (2020). "Semantic Image Manipulation Using Scene Graphs". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5213–5222 (cit. on p. 4).
- Dhariwal, Prafulla and Alex Nichol (2021a). "Diffusion Models Beat GANs on Image Synthesis". In: *NeurIPS*. URL: <https://arxiv.org/abs/2105.05233> (cit. on pp. 20, 90, 97).
- Dhariwal, Prafulla and Alex Nichol (2021b). "Diffusion models beat gans on image synthesis". In: *arXiv preprint arXiv:2105.05233* (cit. on p. 73).
- Ding, Ming, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. (2021). "CogView: Mastering Text-to-Image Generation via Transformers". In: *arXiv preprint arXiv:2105.13290* (cit. on p. 14).
- Donahue, J., P. Krähenbühl, and T. Darrell (2017). "Adversarial Feature Learning". In: *ICLR* (cit. on p. 43).

- Dong, Hanze, Wei Xiong, Deepanshu Goyal, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang (2023). “RAFT: Reward rAnked FineTuning for Generative Foundation Model Alignment”. In: *arXiv preprint arXiv:2304.06767* (cit. on p. 112).
- Dong, Yinpeng, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li (2018). “Boosting adversarial attacks with momentum”. In: *CVPR* (cit. on p. 46).
- Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. (2020). “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (cit. on p. 10).
- Douillard, Arthur, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord (2022). “Dytox: Transformers for continual learning with dynamic token expansion”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9285–9295 (cit. on p. 131).
- Dumoulin, V., I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. Courville (2017). “Adversarially Learned Inference”. In: *ICLR* (cit. on p. 43).
- Engilberge, Martin, Louis Chevallier, Patrick Pérez, and Matthieu Cord (2018). “Finding beans in burgers: Deep semantic-visual embedding with localization”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3984–3993 (cit. on p. 10).
- Epstein, Dave, Taesung Park, Richard Zhang, Eli Shechtman, and Alexei A Efros (2022). “Blobgan: Spatially disentangled scene representations”. In: *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XV*. Springer, pp. 616–635 (cit. on p. 111).
- Esser, Patrick, Robin Rombach, and B. Ommer (2021a). “Taming Transformers for High-Resolution Image Synthesis”. In: *CVPR* (cit. on p. 91).
- Esser, Patrick, Robin Rombach, and Bjorn Ommer (2021b). “Taming transformers for high-resolution image synthesis”. In: *CVPR* (cit. on pp. 14, 15, 44, 73, 110).
- Etienne, Hubert (Nov. 2021). “The future of online trust (and why Deepfake is advancing it)”. In: *AI and Ethics* 1 (cit. on p. 111).
- Faghri, Fartash, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler (2018). “VSE++: Improving Visual-Semantic Embeddings with Hard Negatives”. In: *Proceedings of the British Machine Vision Conference (BMVC)*. URL: <https://github.com/fartashf/vsepp> (cit. on p. 10).
- Feng, Fangxiaoyu, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang (2020). “Language-agnostic bert sentence embedding”. In: *arXiv preprint arXiv:2007.01852* (cit. on pp. 24, 29).
- Feng, Weixi, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang (2022). “Training-

- Free Structured Diffusion Guidance for Compositional Text-to-Image Synthesis". In: *arXiv preprint arXiv:2212.05032* (cit. on p. 92).
- Fernandez, Pierre, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon (2023). "The Stable Signature: Rooting Watermarks in Latent Diffusion Models". In: *arXiv preprint arXiv:2303.15435* (cit. on p. 133).
- Gafni, Oran, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman (2022a). "Make-A-Scene: Scene-Based Text-to-Image Generation with Human Priors". In: *ECCV* (cit. on p. 15).
- Gafni, Oran, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman (2022b). "Make-A-Scene: Scene-Based Text-to-Image Generation with Human Priors". In: *ECCV* (cit. on p. 91).
- Gafni, Oran and Lior Wolf (2020). "Wish you were here: Context-aware human generation". In: *CVPR* (cit. on p. 4).
- Goetschalckx, Lore, Alex Andonian, Aude Oliva, and Phillip Isola (2019). "GANalyze: Toward Visual Definitions of Cognitive Image Properties". In: *arXiv preprint arXiv:1906.10112* (cit. on p. 43).
- Gonzalez-Garcia, Abel, Joost Van De Weijer, and Yoshua Bengio (2018). "Image-to-image translation for cross-domain disentanglement". In: *Advances in neural information processing systems* 31 (cit. on p. 18).
- Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). "Generative adversarial networks". In: *Communications of the ACM* 63.11, pp. 139–144 (cit. on p. 12).
- Goyal, Yash, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh (2017). "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*, pp. 6904–6913 (cit. on p. 10).
- Grechka, Asya, Matthieu Cord, and Jean-Francois Goudou (2021a). "MAGECally invert images for realistic editing". In: *BMVC* (cit. on p. 43).
- Grechka, Asya, Jean-François Goudou, and Matthieu Cord (2021b). "MAGECally invert images for realistic editing". In: *BMVC* (cit. on p. 18).
- Grigorescu, Sorin, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu (2020). "A survey of deep learning techniques for autonomous driving". In: *Journal of Field Robotics* 37.3, pp. 362–386 (cit. on p. 1).
- Guo, Xiaoxiao, Hui Wu, Yupeng Gao, Steven Rennie, and Rogerio Feris (2019). "Fashion IQ: A New Dataset towards Retrieving Images by Natural Language Feedback". In: *arXiv preprint arXiv:1905.12794* (cit. on p. 25).
- Han, Xintong, Zuxuan Wu, Phoenix X Huang, Xiao Zhang, Menglong Zhu, Yuan Li, Yang Zhao, and Larry S Davis (2017). "Automatic spatially-aware fashion concept discovery". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1463–1471 (cit. on p. 25).

- Härkönen, Erik, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris (2020). "GANSpace: Discovering interpretable GAN controls". In: *NeurIPS* (cit. on pp. 18, 43).
- He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015). "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034 (cit. on p. 9).
- He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep residual learning for image recognition". In: *CVPR* (cit. on p. 9).
- Hertz, Amir, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or (2022). "Prompt-to-Prompt Image Editing with Cross Attention Control". In: *arXiv preprint arXiv:2208.01626* (cit. on pp. 67, 73, 92).
- Hessel, Jack, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi (2021). "CLIPScore: A reference-free evaluation metric for image captioning". In: *arXiv preprint arXiv:2104.08718* (cit. on p. 78).
- Heusel, Martin, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter (2017a). "GANs trained by a two time-scale update rule converge to a local Nash equilibrium". In: *NeurIPS* (cit. on p. 48).
- Heusel, Martin, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter (2017b). "GANs trained by a two time-scale update rule converge to a local Nash equilibrium". In: *NeurIPS* (cit. on pp. 74, 99).
- Ho, Jonathan, Ajay Jain, and Pieter Abbeel (2020). "Denoising diffusion probabilistic models". In: *NeurIPS* (cit. on pp. 12, 67).
- Ho, Jonathan and Tim Salimans (2022). "Classifier-free diffusion guidance". In: *arXiv preprint arXiv:2207.12598* (cit. on p. 73).
- Hu, Hexiang, Ishan Misra, and Laurens van der Maaten (2019). "Evaluating text-to-image matching using binary image selection (BISON)". In: *ICCV Workshop on closing the loop between vision and language* (cit. on pp. 81, 139).
- Isola, Phillip, Joseph J Lim, and Edward H Adelson (2015). "Discovering states and transformations in image collections". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1383–1391 (cit. on p. 25).
- Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros (2017). "Image-to-Image Translation with Conditional Adversarial Networks". In: *CVPR* (cit. on pp. 3, 43, 91).
- Jia, Chao, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig (2021). "Scaling up visual and vision-language representation learning with noisy text supervision". In: *arXiv preprint arXiv:2102.05918* (cit. on pp. 24, 30, 45).
- Jing, Yongcheng, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song (2019). "Neural style transfer: A review". In: *Transactions on visualization and computer graphics* 26.11, pp. 3365–3385 (cit. on p. 4).

- Johnson, J., A. Alahi, and L. Fei-Fei (2016). “Perceptual Losses for Real-Time Style Transfer and Super-Resolution”. In: *ECCV* (cit. on p. 4).
- Johnson, Justin, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick (2017). “Clevr: A diagnostic dataset for compositional language and elementary visual reasoning”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2901–2910 (cit. on p. 25).
- Jumper, John, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. (2021). “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596.7873, pp. 583–589 (cit. on p. 1).
- Kang, Minguk, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park (2023). “Scaling up GANs for Text-to-Image Synthesis”. In: *arXiv preprint arXiv:2303.05511* (cit. on p. 12).
- Karpathy, Andrej and Li Fei-Fei (2015). “Deep visual-semantic alignments for generating image descriptions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3128–3137 (cit. on p. 10).
- Karras, T., S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila (2020). “Analyzing and Improving the Image Quality of StyleGAN”. In: *CVPR* (cit. on pp. 19, 43, 56, 57, 59).
- Karras, Tero, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila (2021). “Alias-Free Generative Adversarial Networks”. In: *NeurIPS* (cit. on p. 43).
- Karras, Tero, Samuli Laine, and Timo Aila (2019). “A Style-Based Generator Architecture for Generative Adversarial Networks”. In: *CVPR* (cit. on pp. 43, 57).
- Khrulkov, Valentin and Ivan Oseledets (2022). “Understanding DDPM latent codes through optimal transport”. In: *Applied Mathematics Letters* (cit. on p. 142).
- Kiela, Douwe, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine (2020). “The hateful memes challenge: Detecting hate speech in multimodal memes”. In: *Advances in Neural Information Processing Systems* 33, pp. 2611–2624 (cit. on p. 1).
- Kim, Chung-Il, Meejoung Kim, Seungwon Jung, and Eenjun Hwang (2020). “Simplified Fréchet distance for generative adversarial Nets”. In: *Sensors* 20.6, p. 1548 (cit. on p. 48).
- Kim, Gwanghyun and Jong Chul Ye (2021). “DiffusionCLIP: Text-guided Image Manipulation Using Diffusion Models”. In: *arXiv preprint arXiv:2110.02711* (cit. on p. 67).

- Kim, Wonjae, Bokyung Son, and Ildoo Kim (2021). “Vilt: Vision-and-language transformer without convolution or region supervision”. In: *International Conference on Machine Learning*. PMLR, pp. 5583–5594 (cit. on p. 10).
- Kingma, D. and M. Welling (2014). “Auto-Encoding Variational Bayes”. In: *ICLR* (cit. on p. 44).
- Krause, Jonathan, Michael Stark, Jia Deng, and Li Fei-Fei (2013). “3D Object Representations for Fine-Grained Categorization”. In: *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*. Sydney, Australia (cit. on p. 50).
- Krishna, Ranjay, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. (2017). “Visual genome: Connecting language and vision using crowdsourced dense image annotations”. In: *International journal of computer vision* 123.1, pp. 32–73 (cit. on pp. 25, 27).
- Krizhevsky, A., I. Sutskever, and G. Hinton (2012). “ImageNet Classification with Deep Convolutional Neural Networks”. In: *NeurIPS* (cit. on p. 49).
- Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). “Imagenet classification with deep convolutional neural networks”. In: *Communications of the ACM* 60.6, pp. 84–90 (cit. on pp. 2, 9).
- Lavenant, Hugo and Filippo Santambrogio (2022). “The flow map of the Fokker-Planck equation does not provide optimal transport”. In: *Applied Mathematics Letters* (cit. on p. 142).
- LeCun, Yann, Yoshua Bengio, et al. (1995). “Convolutional networks for images, speech, and time series”. In: *The handbook of brain theory and neural networks* 3361.10, p. 1995 (cit. on p. 9).
- Ledig, Christian, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi (2017). “Photo-realistic single image super-resolution using a generative adversarial network”. In: *CVPR* (cit. on p. 4).
- Lee, Kimin, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu (2023). “Aligning text-to-image models using human feedback”. In: *arXiv preprint arXiv:2302.12192* (cit. on p. 112).
- Lezama, José, Huiwen Chang, Lu Jiang, and Irfan Essa (2022). “Improved Masked Image Generation with Token-Critic”. In: *ECCV*. URL: <https://arxiv.org/abs/2209.04439> (cit. on p. 91).
- Li, Alexander C., Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak (2023). *Your Diffusion Model is Secretly a Zero-Shot Classifier*. arXiv: 2303.16203 [cs.LG] (cit. on p. 111).
- Li, Bowen, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr (2020a). “Manigan: Text-guided image manipulation”. In: *Proceedings of the IEEE/CVF Confer-*

- ence on Computer Vision and Pattern Recognition, pp. 7880–7889 (cit. on pp. 18, 43, 48, 52, 65).
- Li, Bowen, Xiaojuan Qi, Philip HS Torr, and Thomas Lukasiewicz (2020b). “Image-to-image translation with text guidance”. In: *arXiv preprint arXiv:2002.05235* (cit. on p. 18).
- Li, Xiujun, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. (2020). “Oscar: Object-semantics aligned pre-training for vision-language tasks”. In: *European Conference on Computer Vision*. Springer, pp. 121–137 (cit. on pp. 10, 25, 26).
- Li, Yuheng, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee (2023). “GLIGEN: Open-Set Grounded Text-to-Image Generation”. In: *arXiv preprint arXiv:2301.07093* (cit. on p. 92).
- Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick (2014a). “Microsoft COCO: Common objects in context”. In: *ECCV* (cit. on pp. 73, 81).
- Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick (2014b). “Microsoft coco: Common objects in context”. In: *European conference on computer vision*. Springer, pp. 740–755 (cit. on p. 30).
- Logeswaran, Lajanugen and Honglak Lee (2018). “An efficient framework for learning sentence representations”. In: *arXiv preprint arXiv:1803.02893* (cit. on pp. 23, 29).
- Lüddecke, Timo and Alexander S. Ecker (2022). “Image Segmentation Using Text and Image Prompts”. In: *CVPR*. URL: <https://arxiv.org/abs/2112.10003> (cit. on p. 92).
- Lugmayr, Andreas, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool (2022). “RePaint: Inpainting using denoising diffusion probabilistic models”. In: *CVPR* (cit. on pp. 65, 67).
- Ma, Shuang, Jianlong Fu, Chang Wen Chen, and Tao Mei (2018). “Da-gan: Instance-level image translation by deep attention generative adversarial networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5657–5666 (cit. on p. 18).
- Meng, Chenlin, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon (2021). “SDEdit: Guided image synthesis and editing with stochastic differential equations”. In: *ICLR* (cit. on pp. 65, 67, 69, 71, 73, 76, 140).
- Meng, Chenlin, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon (2022). “SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations”. In: *ICLR* (cit. on p. 19).

- Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (cit. on p. 23).
- Mo, Sangwoo, Minsu Cho, and Jinwoo Shin (2018). "Instagan: Instance-aware image-to-image translation". In: *arXiv preprint arXiv:1812.10889* (cit. on p. 18).
- Mou, Chong, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie (2023). "T2I-Adapter: Learning Adapters to Dig out More Controllable Ability for Text-to-Image Diffusion Models". In: *arXiv preprint arXiv:2302.08453*. URL: <https://arxiv.org/abs/2302.08453> (cit. on pp. 92, 98, 99).
- Nasser, Ibrahim M and Samy S Abu-Naser (2019). "Lung cancer detection using artificial neural network". In: *International Journal of Engineering and Information Systems (IJEAIS)* 3.3, pp. 17–23 (cit. on p. 1).
- Nichol, Alex, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen (2021). "Glide: Towards photo-realistic image generation and editing with text-guided diffusion models". In: *arXiv preprint arXiv:2112.10741* (cit. on pp. 65, 66, 77).
- Nichol, Alex, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen (2022). "GLIDE: Towards Photo-realistic Image Generation and Editing with Text-Guided Diffusion Models". In: *ICML*. URL: <https://arxiv.org/abs/2112.10741> (cit. on pp. 91, 100).
- Oord, A. van den, O. Vinyals, and K. Kavukcuoglu (2017). "Neural Discrete Representation Learning". In: *NeurIPS*. URL: <https://papers.nips.cc/paper/2017/file/7a98af17e63a0ac09ce2e96d03992fbc-Paper.pdf> (cit. on pp. 14, 44, 91).
- Oord, Aaron van den, Yazhe Li, and Oriol Vinyals (2018). "Representation learning with contrastive predictive coding". In: *arXiv preprint arXiv:1807.03748* (cit. on p. 10).
- Ouyang, Long, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. (2022). "Training language models to follow instructions with human feedback". In: *Advances in Neural Information Processing Systems* 35, pp. 27730–27744 (cit. on p. 112).
- Paga, Arantxa Casanova, Marlene Careil, Adriana Romero Soriano, Christopher J. Pal, Jakob Verbeek, and Michal Drozdal (2022). "Controllable Image Generation via Collage Representations". In: *ICLR submission* (cit. on p. 89).
- Park, T., M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu (2019). "Semantic Image Synthesis With Spatially-Adaptive Normalization". In: *CVPR*. URL: <https://arxiv.org/pdf/1903.07291.pdf> (cit. on pp. 89, 91, 99).

- Park, Taesung, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu (2019). "Semantic image synthesis with spatially-adaptive normalization". In: *CVPR* (cit. on p. 43).
- Parmar, Gaurav, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu (2023). "Zero-shot Image-to-Image Translation". In: *arXiv preprint arXiv:2302.03027* (cit. on p. 92).
- Parmar, Gaurav, Richard Zhang, and Jun-Yan Zhu (2022). "On Aliased Resizing and Surprising Subtleties in GAN Evaluation". In: *CVPR* (cit. on p. 99).
- Patashnik, Or, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski (2021). "StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery". In: *arXiv preprint arXiv:2103.17249* (cit. on pp. 18, 19, 43, 48, 52, 65).
- Perarnau, Guim, Joost Van De Weijer, Bogdan Raducanu, and Jose M Álvarez (2016). "Invertible conditional gans for image editing". In: *arXiv preprint arXiv:1611.06355* (cit. on p. 12).
- Qi, Chenyang, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen (2023). "FateZero: Fusing Attentions for Zero-shot Text-based Video Editing". In: *arXiv preprint arXiv:2303.09535* (cit. on p. 111).
- Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. (2021a). "Learning transferable visual models from natural language supervision". In: *arXiv preprint arXiv:2103.00020* (cit. on pp. 11, 17, 24, 109).
- Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever (2021b). "Learning Transferable Visual Models From Natural Language Supervision". In: *ICML*. URL: <https://proceedings.mlr.press/v139/radford21a.html> (cit. on pp. 41, 44, 55, 92).
- Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu (2020). "Exploring the limits of transfer learning with a unified text-to-text transformer". In: *The Journal of Machine Learning Research* 21.1, pp. 5485–5551 (cit. on p. 13).
- Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu (2022). "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *JMLR* 21. URL: <https://arxiv.org/abs/1910.10683> (cit. on p. 100).
- Ramesh, Aditya, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen (2022). "Hierarchical text-conditional image generation with CLIP latents". In: *arXiv preprint arXiv:2204.06125* (cit. on p. 12).
- Ramesh, Aditya, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever (2021). "Zero-shot text-to-image generation". In: *ICML* (cit. on pp. 14, 83).

- Razavi, Ali, Aaron van den Oord, and Oriol Vinyals (2019). “Generating Diverse High-Fidelity Images with VQ-VAE-2”. In: *NeurIPS* (cit. on pp. 44, 91).
- Reed, Scott, Zeynep Akata, Xincheng Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee (2016). “Generative adversarial text to image synthesis”. In: *International conference on machine learning*. PMLR, pp. 1060–1069 (cit. on p. 12).
- Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer (2022a). “High-Resolution Image Synthesis with Latent Diffusion Models”. In: *CVPR*. URL: <https://arxiv.org/abs/2112.10752> (cit. on pp. 15, 100).
- Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer (2022b). “High-resolution image synthesis with latent diffusion models”. In: *CVPR* (cit. on pp. 12, 13, 73).
- Roziere, Baptiste, Marie-Anne Lachaux, Lowik Chanussot, and Guillaume Lample (2020). “Unsupervised translation of programming languages”. In: *Advances in Neural Information Processing Systems* 33, pp. 20601–20611 (cit. on p. 1).
- Ruiz, Nataniel, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman (2022). “DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation”. In: *arXiv preprint arXiv:2208.12242* (cit. on p. 4).
- Saharia, Chitwan, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi (2022a). “Palette: Image-to-image diffusion models”. In: *SIGGRAPH* (cit. on pp. 4, 20).
- Saharia, Chitwan, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. (2022b). “Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding”. In: *arXiv:2205.11487* (cit. on pp. 3, 12, 13, 72, 78).
- Saharia, Chitwan, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi (2022c). “Image super-resolution via iterative refinement”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (cit. on p. 4).
- Sauer, Axel, Katja Schwarz, and Andreas Geiger (2022). “Stylegan-xl: Scaling stylegan to large diverse datasets”. In: *ACM SIGGRAPH 2022 conference proceedings*, pp. 1–10 (cit. on p. 12).
- Schönfeld, Edgar, Vadim Sushko, Dan Zhang, Juergen Gall, Bernt Schiele, and Anna Khoreva (2021). “You Only Need Adversarial Supervision for Semantic Image Synthesis”. In: *ICLR*. URL: <https://openreview.net/forum?id=yvQKLqNE6M> (cit. on pp. 89, 91, 98, 99).
- Schuhmann, Christoph, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komat-

- suzaki (2021). “LAION-400M: Open dataset of CLIP-filtered 400 million image-text pairs”. In: *arXiv preprint arXiv:2111.02114* (cit. on p. 73).
- Shen, Yujun, Jinjin Gu, Xiaoou Tang, and Bolei Zhou (2020a). “Interpreting the latent space of GANs for semantic face editing”. In: *CVPR* (cit. on p. 18).
- Shen, Yujun, Jinjin Gu, Xiaoou Tang, and Bolei Zhou (2020b). “Interpreting the latent space of GANs for semantic face editing”. In: *CVPR* (cit. on p. 43).
- Shi, Jing, Ning Xu, Trung Bui, Franck Dernoncourt, Zheng Wen, and Chenliang Xu (2020). “A Benchmark and Baseline for Language-Driven Image Editing”. In: *Proceedings of the Asian Conference on Computer Vision* (cit. on p. 3).
- Shoshan, Alon, Nadav Bhonker, Igor Kviatkovsky, and Gerard Medioni (2021). “GAN-control: Explicitly controllable GANs”. In: *ICCV* (cit. on p. 18).
- Shukor, Mustafa, Guillaume Couairon, and Matthieu Cord (2022a). “Efficient vision-language pretraining with visual concepts and hierarchical alignment”. In: *arXiv preprint arXiv:2208.13628* (cit. on p. 133).
- Shukor, Mustafa, Guillaume Couairon, Asya Grechka, and Matthieu Cord (2022b). “Transformer decoders with multimodal regularization for cross-modal food retrieval”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4567–4578 (cit. on p. 132).
- Silver, David, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dhharshan Kumaran, Thore Graepel, et al. (2018). “A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play”. In: *Science* 362.6419, pp. 1140–1144 (cit. on p. 1).
- Simonyan, K. and A. Zisserman (2015). “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *ICLR* (cit. on p. 45).
- Singh, Amanpreet, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela (2022). “Flava: A foundational language and vision alignment model”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15638–15650 (cit. on p. 132).
- Sinha, Saumya, Bri-Mathias Hodge, and Claire Monteleoni (2022). “Week-ahead solar irradiance forecasting with deep sequence learning”. In: *Environmental Data Science* 1, e28 (cit. on p. 1).
- Sohl-Dickstein, Jascha, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli (2015). “Deep Unsupervised Learning using Nonequilibrium Thermodynamics”. In: *ICML*. URL: <https://proceedings.mlr.press/v37/sohl-dickstein15.html> (cit. on pp. 20, 93).
- Sohn, Kihyuk (2016a). “Improved deep metric learning with multi-class n-pair loss objective”. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 1857–1865 (cit. on p. 31).

- Sohn, Kihyuk (2016b). “Improved deep metric learning with multi-class n-pair loss objective”. In: *Advances in neural information processing systems* 29 (cit. on p. 10).
- Song, Jiaming, Chenlin Meng, and Stefano Ermon (2021). “Denoising diffusion implicit models”. In: *ICLR* (cit. on pp. 13, 19, 66, 68, 69).
- Song, Yale and Mohammad Soleymani (2019). “Polysemous visual-semantic embedding for cross-modal retrieval”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1979–1988 (cit. on p. 23).
- Song, Yang, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole (2021). “Score-Based Generative Modeling through Stochastic Differential Equations”. In: *ICLR*. URL: <https://arxiv.org/abs/2011.13456> (cit. on pp. 20, 93).
- Szegedy, C., V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna (2016). “Rethinking the inception architecture for computer vision”. In: *CVPR* (cit. on p. 48).
- Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna (2016). “Rethinking the inception architecture for computer vision”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826 (cit. on p. 17).
- Tolan, Jamie, Hung-I Yang, Ben Nosarzewski, Guillaume Couairon, Huy Vo, John Brandt, Justine Spore, Sayantan Majumdar, Daniel Haziza, Janaki Vamaraju, et al. (2023). “Sub-meter resolution canopy height maps using self-supervised learning and a vision transformer trained on Aerial and GEDI Lidar”. In: *arXiv preprint arXiv:2304.07213* (cit. on p. 134).
- Touvron, Hugo, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou (2021). “Training data-efficient image transformers & distillation through attention”. In: *International conference on machine learning*. PMLR, pp. 10347–10357 (cit. on pp. 10, 48).
- Tov, Omer, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or (2021). “Designing an encoder for StyleGAN image manipulation”. In: *ACM Transactions on Graphics (TOG)* 40.4, pp. 1–14 (cit. on pp. 52, 57, 59).
- Van Klompenburg, Thomas, Ayalew Kassahun, and Cagatay Catal (2020). “Crop yield prediction using machine learning: A systematic literature review”. In: *Computers and Electronics in Agriculture* 177, p. 105709 (cit. on p. 1).
- Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin (2017). “Attention Is All You Need”. In: *NeurIPS* (cit. on p. 96).
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). “Attention is all you need”. In: *Advances in neural information processing systems* 30 (cit. on pp. 2, 10, 13).

- Vo, Nam, Lu Jiang, and James Hays (2019a). “Let’s Transfer Transformations of Shared Semantic Representations”. In: *arXiv preprint arXiv:1903.00793* (cit. on pp. 26, 27).
- Vo, Nam, Lu Jiang, Chen Sun, Kevin Murphy, Li-Jia Li, Li Fei-Fei, and James Hays (2019b). “Composing text and image for image retrieval-an empirical odyssey”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6439–6448 (cit. on pp. 23, 25).
- Voynov, Andrey and Artem Babenko (2020). “Unsupervised Discovery of Interpretable Directions in the GAN Latent Space”. In: *ICML* (cit. on p. 43).
- Wang, Jianan, Guansong Lu, Hang Xu, Zhenguo Li, Chunjing Xu, and Yanwei Fu (2022). “ManiTrans: Entity-Level Text-Guided Image Manipulation via Token-wise Semantic Alignment and Generation”. In: *CVPR* (cit. on p. 65).
- Wang, Liwei, Yin Li, and Svetlana Lazebnik (2016). “Learning deep structure-preserving image-text embeddings”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5005–5013 (cit. on p. 10).
- Wang, Tengfei, Ting Zhang, Bo Zhang, Hao Ouyang, Dong Chen, Qifeng Chen, and Fang Wen (2022a). “Pretraining is All You Need for Image-to-Image Translation”. In: *arXiv preprint arXiv:2205.12952* (cit. on pp. 66, 89, 91).
- Wang, Tengfei, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen (2022b). “High-fidelity GAN inversion for image attribute editing”. In: *CVPR* (cit. on p. 18).
- Wang, Ting-Chun, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro (2018). “High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs”. In: *CVPR* (cit. on p. 43).
- Wang, Weilun, Jianmin Bao, Wengang Zhou, Dongdong Chen, Dong Chen, Lu Yuan, and Houqiang Li (2022). “Semantic image synthesis via diffusion models”. In: *arXiv preprint arXiv:2207.00050* (cit. on pp. 89, 91, 98, 99).
- Wang, Yaxing, Luis Herranz, and Joost van de Weijer (2020). “Mix and Match Networks: Cross-Modal Alignment for Zero-Pair Image-to-Image Translation”. In: *IJCV* 128.12, pp. 2849–2872 (cit. on p. 43).
- Yu, Jiahui, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang (2018). “Generative image inpainting with contextual attention”. In: *CVPR* (cit. on p. 4).
- Zhang, Lvmin and Maneesh Agrawala (2023). “Adding Conditional Control to Text-to-Image Diffusion Models”. In: *arXiv preprint arXiv:2302.05543*. URL: <https://arxiv.org/abs/2302.05543> (cit. on p. 92).
- Zhang, R., P. Isola, A. Efros, E. Shechtman, and O. Wang (2018). “The Unreasonable Effectiveness of Deep Features as a Perceptual Metric”. In: *CVPR* (cit. on pp. 17, 45, 49, 74).
- Zheng, Zhedong, Liang Zheng, Michael Garrett, Yi Yang, Mingliang Xu, and Yi-Dong Shen (2020). “Dual-path convolutional image-text embeddings with

- instance loss". In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 16.2, pp. 1–23 (cit. on p. 10).
- Zhu, J.-Y., T. Park, P. Isola, and A. Efros (2017). "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks". In: *ICCV* (cit. on p. 43).
- Zhu, Jiapeng, Yujun Shen, Deli Zhao, and Bolei Zhou (2020). "In-domain GAN inversion for real image editing". In: *ECCV* (cit. on p. 18).
- Zhuang, Peiye, Oluwasanmi O Koyejo, and Alex Schwing (2021). "Enjoy Your Editing: Controllable GANs for Image Editing via Latent Space Navigation". In: *ICLR* (cit. on p. 43).

APPENDIX

A.1 Publications

Main publications

- Guillaume Couairon, Matthijs Douze, Matthieu Cord, and Holger Schwenk (2022a). “Embedding Arithmetic of Multimodal Queries for Image Retrieval”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, O-DRUM workshop*, pp. 4950–4958.
- Guillaume Couairon, Asya Grechka, Jakob Verbeek, Holger Schwenk, and Matthieu Cord (2022b). “Flexit: Towards flexible semantic image translation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18270–18279.
- Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord (2023b). “Diffedit: Diffusion-based semantic image editing with mask guidance”. In: *International Conference in Learning Representations*.
- Guillaume Couairon, Marlene Careil, Matthieu Cord, Stéphane Lathuillère, and Jakob Verbeek (2023a). “ZestGuide”. In: *Under Review*.

Other publications

- Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord (2022). “Dytox: Transformers for continual learning with dynamic token expansion”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9285–9295. *Abstract*: Deep network architectures struggle to continually learn new tasks without forgetting the previous tasks. A recent trend indicates that dynamic architectures based on an expansion of the parameters can reduce catastrophic forgetting efficiently in continual learning. However, existing approaches often require a task identifier at test-time, need complex tuning to balance the growing number of parameters, and barely share any information across tasks. As a

result, they struggle to scale to a large number of tasks without significant overhead. In this paper, we propose a transformer architecture based on a dedicated encoder/decoder framework. Critically, the encoder and decoder are shared among all tasks. Through a dynamic expansion of special tokens, we specialize each forward of our decoder network on a task distribution. Our strategy scales to a large number of tasks while having negligible memory and time overheads due to strict control of the parameters expansion. Moreover, this efficient strategy doesn't need any hyperparameter tuning to control the network's expansion. Our model reaches excellent results on CIFAR100 and state-of-the-art performances on the large-scale ImageNet100 and ImageNet1000 while having less parameters than concurrent dynamic frameworks.

- Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela (2022). "Flava: A foundational language and vision alignment model". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15638–15650. *Abstract*: State-of-the-art vision and vision-and-language models rely on large-scale visio-linguistic pretraining for obtaining good performance on a variety of downstream tasks. Generally, such models are often either cross-modal (contrastive) or multi-modal (with earlier fusion) but not both; and they often only target specific modalities or tasks. A promising direction would be to use a single holistic universal model, as a "foundation", that targets all modalities at once – a true vision and language foundation model should be good at vision tasks, language tasks, and cross- and multi-modal vision and language tasks. We introduce FLAVA as such a model and demonstrate impressive performance on a wide range of 35 tasks spanning these target modalities.
- Mustafa Shukor, Guillaume Couairon, Asya Grechka, and Matthieu Cord (2022b). "Transformer decoders with multimodal regularization for cross-modal food retrieval". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4567–4578. *Abstract*: Cross-modal image-recipe retrieval has gained significant attention in recent years. Most work focuses on improving cross-modal embeddings using unimodal encoders, that allow for efficient retrieval in large-scale databases, leaving aside cross-attention between modalities which is more computationally expensive. We propose a new retrieval framework, T-Food (Transformer Decoders with MultiModal Regularization for Cross-Modal Food Retrieval) that exploits the interaction between modalities in a novel regularization scheme, while using only unimodal encoders at test time for efficient retrieval. We also

capture the intra-dependencies between recipe entities with a dedicated recipe encoder, and propose new variants of triplet losses with dynamic margins that adapt to the difficulty of the task. Finally, we leverage the power of the recent Vision and Language Pretraining (VLP) models such as CLIP for the image encoder. Our approach outperforms existing approaches by a large margin on the Recipe1M dataset. Specifically, we achieve absolute improvements of 8.1% (72.6 R@1) and +10.9% (44.6 R@1) on the 1k and 10k test sets respectively.

- Mustafa Shukor, Guillaume Couairon, and Matthieu Cord (2022a). “Efficient vision-language pretraining with visual concepts and hierarchical alignment”. In: *arXiv preprint arXiv:2208.13628*. *Abstract*: Vision and Language Pretraining has become the prevalent approach for tackling multimodal downstream tasks. The current trend is to move towards ever larger models and pretraining datasets. This computational headlong rush does not seem reasonable in the long term to move toward sustainable solutions, and de facto excludes academic laboratories with limited resources. In this work, we propose a new framework, dubbed ViCHA, that efficiently exploits the input data to boost the learning by: (a) a new hierarchical cross-modal alignment loss, (b) new self-supervised scheme based on masked image modeling, (c) leveraging image-level annotations, called Visual Concepts, obtained with existing foundation models such as CLIP to boost the performance of the image encoder. Although pretrained on four times less data, our ViCHA strategy outperforms other approaches on several downstream tasks such as Image-Text Retrieval, VQA, Visual Reasoning, Visual Entailment and Visual Grounding.
- Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon (2023). “The Stable Signature: Rooting Watermarks in Latent Diffusion Models”. In: *arXiv preprint arXiv:2303.15435*. *Abstract*: Generative image modeling enables a wide range of applications but raises ethical concerns about responsible deployment. This paper introduces an active strategy combining image watermarking and Latent Diffusion Models. The goal is for all generated images to conceal an invisible watermark allowing for future detection and/or identification. The method quickly fine-tunes the latent decoder of the image generator, conditioned on a binary signature. A pre-trained watermark extractor recovers the hidden signature from any generated image and a statistical test then determines whether it comes from the generative model. We evaluate the invisibility and robustness of the watermarks on a variety of generation tasks, showing that Stable Signature works even after the images are modified. For instance, it detects the origin

of an image generated from a text prompt, then cropped to keep 10% of the content, with 90+% accuracy at a false positive rate below 10^{-6} .

- Jamie Tolan, Hung-I Yang, Ben Nosarzewski, Guillaume Couairon, Huy Vo, John Brandt, Justine Spore, Sayantan Majumdar, Daniel Haziza, Janaki Vamaraju, et al. (2023). “Sub-meter resolution canopy height maps using self-supervised learning and a vision transformer trained on Aerial and GEDI Lidar”. In: *arXiv preprint arXiv:2304.07213 Abstract*: Vegetation structure mapping is critical for understanding the global carbon cycle and monitoring nature-based approaches to climate adaptation and mitigation. Repeat measurements of these data allow for the observation of deforestation or degradation of existing forests, natural forest regeneration, and the implementation of sustainable agricultural practices like agroforestry. Assessments of tree canopy height and crown projected area at a high spatial resolution are also important for monitoring carbon fluxes and assessing tree-based land uses, since forest structures can be highly spatially heterogeneous, especially in agroforestry systems. Very high resolution satellite imagery (less than one meter (1m) ground sample distance) makes it possible to extract information at the tree level while allowing monitoring at a very large scale. This paper presents the first high-resolution canopy height map concurrently produced for multiple sub-national jurisdictions. Specifically, we produce canopy height maps for the states of California and São Paulo, at sub-meter resolution, a significant improvement over the ten meter (10m) resolution of previous Sentinel / GEDI based worldwide maps of canopy height. The maps are generated by applying a vision transformer to features extracted from a self-supervised model in Maxar imagery from 2017 to 2020, and are trained against aerial lidar and GEDI observations. We evaluate the proposed maps with set-aside validation lidar data as well as by comparing with other remotely sensed maps and field-collected data, and find our model produces an average Mean Absolute Error (MAE) within set-aside validation areas of 3.0 meters.

A.2 ImageNet transformations dataset

To evaluate Semantic Image Editing, we have designed the ImageNet transformations dataset, used in chapters 4 and 5. To design transformation queries from ImageNet classes, we have grouped classes into clusters by semantic similarity, upon manual inspection of the WordNet hierarchy of classes. The resulting clusters are shown in Table A.1. This process resulted in 273 classes gathered in 47 clusters. We have not included all ImageNet classes because (i) we wanted to reduce the large number of dog breed classes, and (ii) a lot of classes were “standalone classes” with no natural target for transformation among the other classes. The clusters are then grouped into 13 bigger “groups”.

A.2.1 FlexIT Ablation results

In this section, we give the detailed evaluation scores for the FLEXIT method (Chapter 4) with different configurations. We also show quantitative results for additional ablation experiments.

In Table A.2, we show quantitative results for the main configuration parameters. In Table A.3, we show ablations for combining multiple CLIP networks and using multiple data augmentations in the multimodal encoder. We also report the runtime needed for each algorithm.

Group	Cluster	Classes
bird	bird of prey	bald eagle, kite, great grey owl
bird	finch	indigo bunting, goldfinch, house finch, junco
bird	grouse	black grouse, prairie chicken, ptarmigan, ruffed grouse
bird	seabird	king penguin, albatross, pelican, European gallinule, black swan
bird	wading bird	goose, oystercatcher, little blue heron, black stork, bustard, flamingo, spoonbill
container	bag	backpack, plastic bag, purse
container	food container	water jug, beer bottle, water bottle, wine bottle, coffee mug, vase, coffeepot, teapot, measuring cup, cocktail shaker
device	electronics	cassette player, cellular telephone, computer keyboard, desktop computer, dial telephone, hard disc, iPod, laptop
device	measuring	analog clock, digital clock, wall clock, stopwatch, digital watch, odometer, barometer
dog	hound	English foxhound, Italian greyhound, Afghan hound, basset, beagle, otterhound
dog	sporting dog	English springer, cocker spaniel, golden retriever, Irish setter
dog	terrier	American Staffordshire terrier, wire-haired fox terrier, standard schnauzer, Border terrier, Irish terrier, Yorkshire terrier
dog	toy dog	papillon, Chihuahua, Japanese spaniel, Shih-Tzu, toy terrier
dog	working dog	collie, German shepherd, Rottweiler, miniature pinscher, French bulldog, Siberian husky, boxer, Eskimo dog
edible	edible fruit	Granny Smith, strawberry, lemon, orange, banana, custard apple, fig, pineapple, pomegranate
edible	sandwich	cheeseburger, hotdog, bagel
edible	vegetable	bell pepper, broccoli, cauliflower, spaghetti squash, zucchini, butternut squash, artichoke, cardoon, cucumber
fungus	fungus	bolete, coral fungus, earthstar, gyromitra, hen-of-the-woods, stinkhorn
insect	beetle	ground beetle, ladybug, leaf beetle, long-horned beetle, tiger beetle, weevil
insect	butterfly	monarch, admiral, cabbage butterfly, lycaenid, ringlet, sulphur butterfly
insect	spider	black widow, garden spider, tarantula, wolf spider, scorpion
mammal	bear	American black bear, brown bear, ice bear, sloth bear, giant panda, lesser panda
mammal	bovid	ox, ibex, bighorn, gazelle, impala, water buffalo, ram, bison
mammal	canine	Arctic fox, grey fox, red fox, African hunting dog, dingo, coyote, red wolf, timber wolf, white wolf, hyena
mammal	equine	sorrel, zebra
mammal	feline	Persian cat, tabby, cheetah, jaguar, leopard, lion, snow leopard, tiger
mammal	great ape	chimpanzee, gorilla, orangutan
mammal	monkey	capuchin, spider monkey, squirrel monkey, baboon, guenon, macaque
music. instr.	percussion	chime, drum, gong, maraca, marimba, steel drum
music. instr.	stringed	cello, violin, acoustic guitar, electric guitar, banjo
music. instr.	wind	bassoon, oboe, sax, flute, cornet, French horn, trombone
object	ball	golf ball, ping-pong ball, rugby ball, soccer ball, tennis ball
object	handtool	hammer, plane, plunger, screwdriver, shovel
object	headdress	bathing cap, shower cap, bonnet, cowboy hat, sombrero, football helmet
reptile	amphibian	bullfrog, tree frog, axolotl, spotted salamander, common newt, eft, European fire salamander
reptile	snake	rock python, boa constrictor, green mamba, Indian cobra, diamondback, sidewinder, horned viper, king snake, green snake, thunder snake
reptile	turtle	box turtle, mud turtle, terrapin
sea life	aqu. mammal	killer whale, grey whale, sea lion, dugong
sea life	bony fish	goldfish, tench, eel, anemone fish, lionfish, gar, sturgeon
sea-life	crab	American lobster, Dungeness crab, fiddler crab, king crab, rock crab, crayfish, hermit crab, isopod
sea life	shark	great white shark, tiger shark, hammerhead
vehicle	bicycle	motor scooter, tricycle, unicycle, mountain bike, moped
vehicle	boat	speedboat, lifeboat, canoe, fireboat, gondola
vehicle	car	ambulance, beach wagon, cab, convertible, jeep, limousine, minivan, sports car
vehicle	locomotive	electric locomotive, steam locomotive
vehicle	sailing vessel	catamaran, trimaran, schooner
vehicle	truck	minivan, police van, fire engine, garbage truck, pickup, tow truck, trailer truck, school bus

Table A.1. – Groups and clusters of the ImageNet classes used to define the transformation queries.

	Acc.↑	LPIPS↓	CSFID↓	SFID↓
$\lambda_I = 0$	64.8	27.6	65.4	12.3
$\lambda_I = 0.1$	60.6	25.9	57.8	8.3
$\lambda_I = 0.2$	52.6	24.6	55.9	6.4
$\lambda_I = 0.3$	45.8	23.5	56.3	5.5
$\lambda_I = 0.4$	38.6	22.6	58.6	5.0
$\lambda_S = 0.0$	34.3	23.8	60.2	4.8
$\lambda_S = 0.2$	45.9	24.0	57.3	5.5
$\lambda_S = 0.4$	52.6	24.6	55.9	6.4
$\lambda_S = 0.5$	56.2	25.0	56.5	7.1
$\lambda_S = 0.8$	60.0	26.5	65.5	11.7
$\lambda_z = 0.0$	59.4	26.5	56.1	7.1
$\lambda_z = 0.05$	52.6	24.6	55.9	6.4
$\lambda_z = 0.1$	51.0	23.3	56.7	6.3
$\lambda_p = 0.05$	66.2	28.8	56.0	7.9
$\lambda_p = 0.1$	59.1	26.4	56.0	7.2
$\lambda_p = 0.15$	52.6	24.6	55.9	6.4
$\lambda_p = 0.2$	47.9	23.3	57.5	6.3
ℓ_1	54.2	24.6	56.3	6.5
ℓ_2	52.4	24.5	55.9	6.8
$\ell_{2,1}$	52.6	24.6	55.9	6.4
$lr = 0.025$	47.6	22.5	58.3	6.0
$lr = 0.5$	52.6	24.6	55.9	6.4
$lr = 0.1$	60.4	27.6	54.8	7.2
resolution 256	53.8	24.8	56.8	7.2
resolution 288	52.6	24.6	55.9	6.4
resolution 320	54.3	24.0	57.4	7.3

Table A.2. – FLEXIT ablation results. lr is the learning rate. Lines corresponding to our default configuration are marked in light grey. The norms ℓ_1 , ℓ_2 , and $\ell_{2,1}$ refer to the distance used for regularization in the VQGAN latent space. Best values for each metric are shown in bold inside each group of parameter values.

networks	d	Acc. \uparrow	LPIPS \downarrow	CSFID \downarrow	SFID \downarrow	sec. /im
ViT-B/32	0	9.4	21.8	92.7	7.4	27s
ViT-B/32	1	37.5	26.4	76.5	11.1	27s
ViT-B/32	8	35.1	25.4	76.9	10.7	33s
ViT-B/32	32	35.5	25.0	77.7	10.8	53s
RN50x4	0	13.4	23.8	91.6	11.8	35s
RN50x4	1	32.5	27.4	80.2	13.7	35s
RN50x4	8	31.0	25.2	77.3	12.3	53s
RN50x4	32	27.0	24.2	79.1	11.7	122s
2 nets	0	23.0	22.8	80.7	9.5	39s
2 nets	1	50.6	26.4	63.2	8.9	39s
2 nets	8	47.8	24.9	62.7	8.4	64s
2 nets	32	47.4	24.2	62.9	8.1	160s
3 nets	0	30.4	22.5	72.2	8.3	45s
3 nets	1	54.9	26.0	56.7	6.7	45s
3 nets	8	52.6	24.6	55.9	6.4	75s
3 nets	32	51.7	24.0	56.7	6.7	190s
5 nets	0	39.6	22.4	66.8	7.7	70s
5 nets	1	60.3	25.5	51.9	5.5	70s
5 nets	8	60.1	23.9	52.1	5.4	176s
5 nets	32	52.0	22.8	52.7	5.2	560s

Table A.3. – Ablation results for the multimodal encoder components. d is the number of augmentations. $d = 0$ means that the encoder takes the unchanged image as input; For $d = 1$, the encoder takes only one (augmented image), which explains why the edit time is the same as $d = 0$. When considering n CLIP networks, we take the first n elements in the following list: RN50x4, ViT-B/32, RN50, ViT-B/16, RN50x16. Our default configuration is marked in light grey. Last column gives computation time per image in seconds.

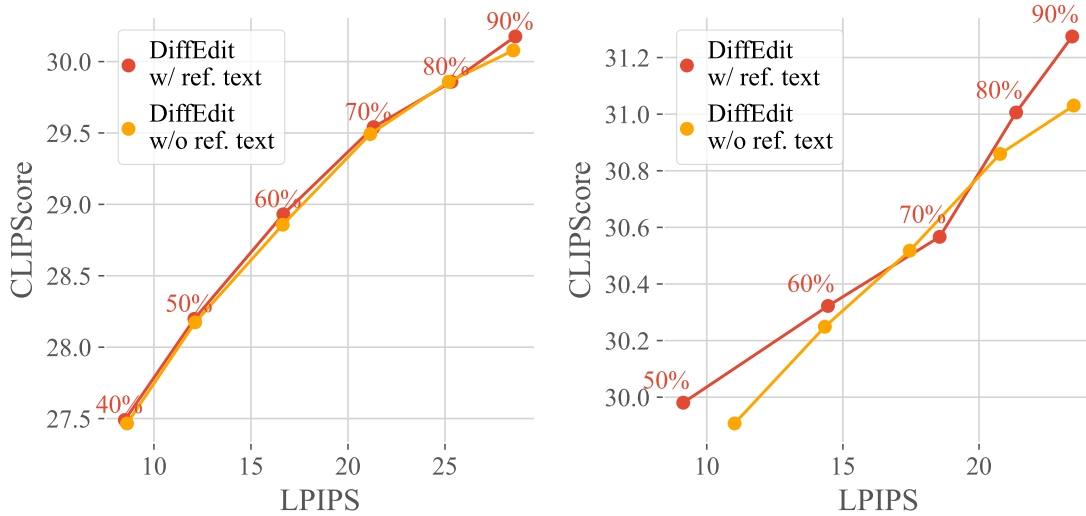


Figure A.1. – Results on COCO: unfiltered (left) and filtered (right). While having a small impact overall, for the filtered set using the reference text is beneficial, especially at high encoding ratios, e.g. 90%.

A.3 DiffEdit Experiments on filtered COCO dataset

In this section, we investigate why there is little difference between using or not the reference text to compute the mask on our COCO queries. In Figure 5.12 we show several editing queries on the COCO dataset taken from the BISON dataset (Hexiang Hu et al. 2019). Generally, the text query describes a scene similar to the one in the input image, and it is possible to match the text query by editing only a fraction of the input image. However, we find that while queries have been built to be close to a caption of the input image, most of the time the query is not well aligned with the caption. We create a filtered version of this dataset, for which queries are structurally similar to the caption, i.e. where only a few words are changed, but the grammatical structure stays the same. We use the filtering criterion that the total number of words inserted/deleted/replaced must not exceed 25% of the total number of words in the original caption, resulting in a total of 272 queries out of 50k original queries. In Figure A.1 we compare results with and without filtering, and observe that for the images with small caption edits the gain of DIFFEDIT (w/ ref. text) compared to *Encode-Decode* is somewhat larger than on the unfiltered dataset. Moreover, using the original caption as reference text to compute the mask gives higher CLIPScore, especially at high encoding ratio. This illustrates that a well chosen reference text helps to generate better editing masks.

A.4 Theoretical results for DiffEdit

In this section, we prove the bounds given in Chapter 5. We reused notations from Proposition 5.1 in the main paper. We also discuss links to optimal transport.

A.4.1 Proof of SDEdit bound

Proposition A.1. *Suppose that $\|\epsilon_\theta(\mathbf{x}, Q, t)\|_2 \leq C$ for all $\mathbf{x} \in \mathcal{X}, t \in [0, 1]$. Then*

$$\mathbb{E}_{\substack{(\mathbf{x}_0, Q) \sim p_D \\ \epsilon \sim \mathcal{N}(0, 1)}} \|\mathbf{x}_0 - D_r(G_r(\mathbf{x}_0, \epsilon), Q)\|_2 \leq (C + 1)\tau \quad (\text{A.1})$$

Proof:

Let $T, \mathbf{x}_r = G_r(\mathbf{x}_0, \epsilon)$, $\mathbf{y}_r = \mathbf{x}_r$ and $\mathbf{y}_0 = D_r(\mathbf{y}_r, Q)$. Then

$$\left\| \frac{\mathbf{x}_r}{\sqrt{\alpha_r}} - \mathbf{y}_0 \right\| = \left\| \frac{\mathbf{y}_r}{\sqrt{\alpha_r}} - \frac{\mathbf{y}_0}{\sqrt{\alpha_0}} \right\| = \left\| \int_\tau^0 \epsilon_\theta(x_t, Q, t) dt \right\| \leq C\tau. \quad (\text{A.2})$$

Since $\frac{\mathbf{x}_r}{\sqrt{\alpha_r}} = \mathbf{x}_0 + \tau\epsilon$, we have $\|\mathbf{x}_0 - \mathbf{y}_0\| \leq \|\mathbf{x}_0 + \tau\epsilon - \mathbf{y}_0\| + \|\tau\epsilon\| \leq C\tau + \tau$ which concludes the proof.

In the SDEdit paper (Meng et al. 2021), a proof similar to what we state is given, with three main differences: (i) the proof is given in the case of variance-exploding Stochastic Differential Equation (VE-SDE), which needs adaption for our setting which uses variance-preserving SDE; (ii) the bound is derived in the case of a stochastic differential equation, whereas we use a deterministic DDIM process; (iii) the bound is given by controlling the probability tail, whereas we only consider the expectancy of edit distance. However, despite these differences, the spirit of the proof is the same as here.

A.4.2 Proof of proposition 2

Proposition A.2. *Suppose that $\epsilon_\theta(\cdot, Q, t)$ is K_1 -lipschitz and κ_2 defined as*

$$\kappa_2(\mathbf{x}_0) = \max_{t \in [0, 1]} \|\epsilon_\theta(E_t(\mathbf{x}_0), Q, t) - \epsilon_\theta(E_t(\mathbf{x}_0), \emptyset, t)\| \quad (\text{A.3})$$

Let $K_2 = \mathbb{E}_{\mathbf{x}_0} \kappa_2(\mathbf{x}_0)$. Then for all encoding ratio r , with $\tau = \sqrt{\alpha_r^{-1} - 1}$,

$$\mathbb{E}_{\mathbf{x}_0} \|\mathbf{x}_0 - D_r(E_r(\mathbf{x}_0), Q)\| \leq \frac{K_2 \tau}{\sqrt{\tau^2 + 1}} \left(\tau + \sqrt{\tau^2 + 1} \right)^{K_1} \quad (\text{A.4})$$

Proof: Let σ be a time-dependent variable defined as $\sigma(t) = \sqrt{\alpha_t^{-1} - 1}$. Let $\mathbf{u} = \mathbf{x}/\sqrt{\alpha} = \mathbf{x}\sqrt{1 + \sigma^2}$ and $\mathbf{v} = \mathbf{y}\sqrt{1 + \sigma^2}$. \mathbf{u} and \mathbf{v} are solutions of the following differential system:

$$d\mathbf{u}|_t = \epsilon_\theta(\mathbf{u}/\sqrt{1 + \sigma^2}, \emptyset, t) d\sigma, \quad (\text{A.5})$$

$$d\mathbf{v}|_t = \epsilon_\theta(\mathbf{v}/\sqrt{1 + \sigma^2}, Q, t) d\sigma, \quad (\text{A.6})$$

$$\mathbf{u}(r) = \mathbf{v}(r) = E_r(\mathbf{x}_0)\sqrt{1 + \sigma^2}. \quad (\text{A.7})$$

Let $\mathbf{w} = \|\mathbf{u} - \mathbf{v}\|$, then $\mathbf{w}|_{t=r} = 0$ and

$$d\mathbf{w}|_t \leq \|d\mathbf{u}|_t - d\mathbf{v}|_t\| = \|(\epsilon_\theta(\mathbf{x}, \emptyset, t) - \epsilon_\theta(\mathbf{y}, Q, t))d\sigma\| \quad (\text{A.8})$$

$$\leq \|(\epsilon_\theta(\mathbf{x}, \emptyset, t) - \epsilon_\theta(\mathbf{x}, Q, t))d\sigma\| + \|(\epsilon_\theta(\mathbf{x}, Q, t) - \epsilon_\theta(\mathbf{y}, Q, t))d\sigma\| \quad (\text{A.9})$$

$$\leq \kappa_2(\mathbf{x}_0)d\sigma + K_1\|\mathbf{x} - \mathbf{y}\|d\sigma \quad (\text{A.10})$$

$$\leq \left(\kappa_2(\mathbf{x}_0) + \frac{K_1}{\sqrt{1 + \sigma^2}} \mathbf{w} \right) d\sigma. \quad (\text{A.11})$$

By integration we get

$$\mathbf{w}(t) \leq \kappa_2(\mathbf{x}_0) * (\tau - t) + \int_t^\tau \frac{K_1}{\sqrt{1 + \sigma^2}} \mathbf{w}(\sigma) d\sigma.$$

From here we can apply Grönwall's inequality:

$$\mathbf{w}(0) \leq \kappa_2(\mathbf{x}_0)\tau \exp \left(\int_0^\tau \frac{K_1}{\sqrt{1 + s^2}} ds \right) \quad (\text{A.12})$$

$$\leq \kappa_2(\mathbf{x}_0)\tau \exp \left(K_1 \log(\tau + \sqrt{\tau^2 + 1}) \right) \quad (\text{A.13})$$

$$\leq \kappa_2(\mathbf{x}_0)\tau \left(\tau + \sqrt{\tau^2 + 1} \right)^{K_1}. \quad (\text{A.14})$$

Which finally gives

$$\|\mathbf{x}_0 - \mathbf{y}_0\| \leq \frac{\kappa_2(\mathbf{x}_0)\tau}{\sqrt{\tau^2 + 1}} \left(\tau + \sqrt{\tau^2 + 1} \right)^{K_1}. \quad (\text{A.15})$$

Taking the expectation w.r.t. the input image \mathbf{x}_0 gives the final result:

$$\mathbb{E}_{\mathbf{x}_0} \|\mathbf{x}_0 - D_T(E_T(\mathbf{x}_0), Q)\| \leq \frac{K_2\tau}{\sqrt{\tau^2 + 1}} \left(\tau + \sqrt{\tau^2 + 1} \right)^{K_1} \quad (\text{A.16})$$

which concludes the proof.

A.4.3 Links to optimal transport theory

The reverse DDIM encoder E_r maps the distribution of images $p_0 = p_D$ to the distribution p_r of images noised at timestep r . Khrulkov and Oseledets 2022 suggested that E_r could be an optimal transport map between p_0 and p_r , minimizing the transport cost $\mathbb{E}_{\mathbf{x}_0} \|\mathbf{x}_0 - E_r(\mathbf{x}_0)\|_2^2$. This means that the encoded images are, on average, as close as possible to the input images, while following the correct distribution p_r . It would entail that the unconditional decoder $D_r = E_r^{-1}$ would be an optimal transport map between p_r and p_0 , and moreover that the conditional decoder $D_r(\cdot, Q)$ would be an optimal transport map between the distributions $p_r(\cdot|Q)$ and $p_0(\cdot|Q)$ conditioned by text description Q . Under the hypothesis that p_r is very close to $p_r(\cdot|Q)$, then the Encode-Decode algorithm would be the combination of two optimal transport maps E_r and $D_r(\cdot, Q)$, mapping p_0 to p_r and then $p_r \simeq p_r(\cdot|Q)$ to $p_0(\cdot|Q)$. This is a very interesting property and we make the connection with the desired properties of semantic image editing, which can be expressed as an optimal transport problem. Given two distribution of images p_1, p_2 (lets say *cats* and *dogs*), the aim is to find the function f that performs the expected edit (changing images of *cats* into images of *dogs*) while minimally editing the image, which can be expressed mathematically as:

$$f = \arg \min_f \mathbb{E}_{\mathbf{x}} \|\mathbf{x} - f(\mathbf{x})\| \quad \text{s.t.} \quad p_2 = f_{\#} p_1, \quad (\text{A.17})$$

where $f_{\#}$ is the push-forward measure. The function $D_r(\cdot, Q) \circ E_r$ is not a solution of this optimal transport problem, because (i) it was proven that the reverse DDIM encoder is not the optimal transport map for some distributions (Lavenant and Santambrogio 2022), and (ii) the composition of two optimal transport maps is not necessarily an optimal transport map. However, experiments and numerical simulations suggest that E_r is very close from an optimal transport map. It would

be interesting to study the “optimality defect” of E_r and of the editing function $D_r(\cdot, Q) \circ E_r$. We leave this for future work.

