Integrating Geometric Understanding in Generative Diffusion Models With Text Instructions

---

A

Thesis

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

---

in partial fulfillment

of the requirements for the degree

Master of Science

by

Ishita Gupta

December 2024

APPROVAL SHEET

This

Thesis

is submitted in partial fulfillment of the requirements
for the degree of

Master of Science

Author: Ishita Gupta

This Thesis has been read and approved by the examing committee:

Advisor: Prof. Miaomiao Zhang

Advisor:

Committee Member: Prof. Tom Fletcher

Committee Member: Prof. Zezhou Cheng

Committee Member:

Committee Member:

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:

Jennifer L. West, School of Engineering and Applied Science

December 2024

Abstract

Generative models have revolutionized image synthesis and editing tasks, achieving unparalleled success in semantic and stylistic transformations. While they have come a long way, their ability to learn and apply precise geometric transformations remains limited by explicit conditioning methods and dependence on substantially labeled data. This limitation restricts their ability to adapt to real-world applications requiring spatial accuracy. This thesis presents a novel approach that enables generative models to implicitly learn and apply geometric transformations, particularly rotation, through latent space manipulation.

An integrated system is proposed that combines text-guided generation with geometric reasoning, equipping generative models with the ability to learn geometric features and integrate geometric reasoning into the learning pipeline. By combining latent space learning with text-based guidance and diffusion-based denoising, the proposed framework achieves precise and interpretable geometric transformations, specifically focusing on rotations as a proof of concept.

The proposed model aims to learn a latent representation that captures the geometric difference between the source and target images. The transformation parameter (rotation angle) is learned directly from the latent space. The latent space, combined with the source image and text embeddings, is refined using a diffusion model, which allows the generative models to implicitly learn and apply geometric transformations. The diffusion model enhances the coherence of the transformed outputs while maintaining consistency with the geometric constraints provided via text prompts.

Experimental results demonstrate that the proposed framework can effectively capture and learn rotation transformations. The proposed model outperforms the baseline methods (FastEdit, SDEdit, and InstructPix2Pix) by achieving an FID score of 4.85, IS of 3.53, and SSIM of 0.88, and shows alignment with ground truth transformations. The modularity of the architecture suggests its potential for generalizing transformations beyond rotations and paves the way for more robust generative modeling techniques. It opens avenues for applications in medical imaging, robotics, and augmented reality, where precise and efficient image manipulation is critical.

Acknowledgements

I would like to express my deepest gratitude to my advisor, Professor Miaomiao Zhang for her support throughout my coursework and research journey at the University of Virginia. Her dedication to her students as an advisor is unmatched, and her mentorship, patience, and insightful feedback have been instrumental in shaping this thesis. I am grateful to have had the opportunity to learn from her during my graduate studies.

I extend my sincere thanks to Professor Tom Fletcher and Professor Zezhou Cheng for kindly agreeing to serve on my thesis committee. I sincerely appreciate their willingness to share their invaluable expertise and guidance, which challenged me to consider my research from different angles and to hold myself to a higher standard. I would also like to acknowledge my research mentor, Tonmoy Hossain for his continuous mentorship and support during my time at the Medical Image Analysis (MIA) lab. His guidance has been integral to my growth as a researcher.

Finally, I am profoundly grateful to my family for their unwavering love, support, and encouragement, especially through my graduate journey. Their belief in my potential is one of my strongest strengths. They have always been there to cheer me on and express their sense of pride in my work. Their excitement over my smallest victories and their reassurance during moments of doubt made a profound difference. I am equally thankful to my friends, who have stood by me and offered both motivation and comfort during challenging times. Whether it was celebrating milestones or navigating the ups and downs of graduate life, their presence has made this journey all the more meaningful.

I am extremely grateful to have such an endearing support group. This thesis is a testament to the collective efforts and encouragement of all those who have supported me along the way and I will carry their impact with me into all my future endeavors.

Table of Contents

List of Tables

List of Figures

Acronyms

MLP Multilayer Perceptron

GAN Generative Adversarial Network

IP2P Instruct Pix2Pix

VAE Variational Autoencoder

LoRA Low-Rank Adaptation

SDE Stochastic Differential Equation

STN Spatial Transformer Network

GAGAN Geometry-Aware Generative Adversarial Network

MSE Mean Squared Error

CLIP Contrastive Language-Image Pre-Training

IS Inception Score

FID Fréchet Inception Distance

SSIM Structural Similarity Index Measure

AR Augmented Reality

VR Virtual Reality

1 Introduction

1.1 Image Generation Models

Equipped with the ability to generate new data samples from a given dataset, generative models are a cornerstone of artificial intelligence. They operate by learning the underlying probability distribution of the dataset to produce outputs that are statistically similar to the original dataset. Such models have been remarkably successful in the domain of image generation, allowing for applications such as image generation and style transfer [1] and [2]. Some prominent types of generative models include:

- Generative Adversarial Networks (GANs): GANs [3] consist of a generator and a discriminator that engage together in a zero-sum game. The generator's goal is to produce images that are indistinguishable from the real images, while the discriminator evaluates them against the actual data to determine if they are real or fake. This setup allows GANs to produce realistic and high-quality images.

- Variational Autoencoders (VAEs): VAEs [4] use a probabilistic approach to encode input data into a latent space and decode it back to reconstruct the data. They impose a probabilistic distribution (usually Gaussian) on the latent variables which helps in generating new data and learning smooth latent representations.

- Diffusion Models: A much more recent development, diffusion models [5] start with a distribution of random noise and gradually learn to reverse this noise to obtain meaningful data samples. Iterative denoising is conducted, guided by a model trained to predict the noise that had been added in each step backward and allow the model to generate detailed images.

1.2 Text-Guided Image Transformation

Text-guided image transformations lie at an innovative intersection of computer vision and natural language processing. Text-guided image-to-image models [6] [7] are used in generative tasks when users want to modify images while keeping certain elements intact, based on textual inputs. These models transform the input image to a new image based on a text prompt, thus combining both the structure of the input image and the semantic guidance of the text as shown in figure 1.

Figure 1: Text-Guided Image Transformation on an image of the Rotunda with IP2P using the prompt "transform into a winter theme"

Here is a high-level overview of the transformation process:

1. The text input is interpreted using natural language processing techniques. This involves understanding the semantics of the text, identifying key descriptors, and mapping these descriptors to visual attributes.

2. The next step depends on the model's architecture. An image is either generated from scratch (as in the case of DALL·E [8]) or the existing image is modified (as in the case of CLIP-guided diffusion models). Visual elements of the image are adjusted by the model - this can range from altering colors or adding objects to the image.

In the context of this work, the CLIP model [9] developed by OpenAI has been utilized. CLIP is a model trained on a vast amount of image and text data to learn visual concepts from natural language descriptions. Its ability to understand and correlate textual and visual information makes it a foundational tool for text-guided image transformations. It is effective at guiding diffusion models to ensure that the transformations align with the textual descriptions provided by the user.

1.3 Success of Current Generative Diffusion Models in Image Editing

Generative models, such as DALL-E 2 [10], Stable Diffusion [11], and InstructPix2Pix [12], have demonstrated remarkable success in generating and editing images. They are capable of generating visually realistic and coherent images by iteratively refining noisy data into meaningful outputs, guided by text prompts. They can handle a variety of tasks, including style transfer and inpainting which allows text-based control of semantic edits like color changes, object addition or removal, or stylistic transformations.

However, despite their success, current diffusion models fall short when it comes to geometric transformations [13] [14].

1.4 Importance of Geometric Transformations

Rotations, scaling, and translations are fundamental operations in image editing [15] [16]. They are used for tasks like aligning objects in an image, correcting distortions in real-world images or scans, and creating visually consistent scenes in augmented or virtual environments. Geometric transformations are also widely used for augmenting datasets in computer vision to improve model generalization for classification, detection, and segmentation tasks.

On a much more domain-level analysis – they are critical in the field of medical imaging to align medical scans to ensure accurate diagnosis. Camera feeds in autonomous vehicles often require real-time geometric adjustments to align objects and detect obstacles. In the domain of AR (Augmented Reality) or VR (Virtual Reality) – immersive environments rely on accurate geometric transformations [17] to position and align virtual objects [18].

Thus, understanding geometric transformations allows users to make precise adjustments through intuitive interfaces, such as text prompts like "rotate by X degrees" or "scale by a factor of 1.5". This is necessary for accessibility so that even non-expert users can interact with generative models seamlessly.

1.5 Challenges in Geometric Transformations with Current Generative Models

From basic preprocessing tasks such as cropping and resizing to advanced operations like 3D modeling – geometric transformations are an important application of computer vision. In the context of generative models, these transformations involve the modification of spatial image attributes like rotation, scaling, and translation, to generate varied perspectives of the same object or scene.

Generative models have been a revolution for machines to understand and generate multimedia content. They excel in semantic and stylistic transformations. However, integrating geometric transformations into generative models presents a series of challenges [13]. As shown in Figure 2, they often struggle with handling such transformations since their focus is on pixel-level data rather than higher-level abstractions like shape and spatial arrangement.

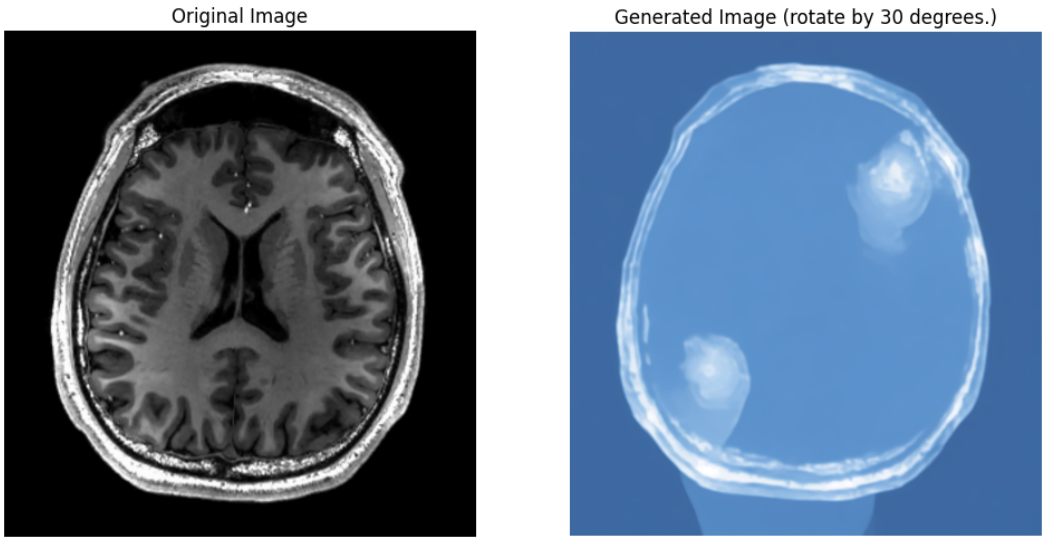

Figure 2: Attempted Text-Guided Rotation Transformation on a brain MRI image using IP2P using the prompt "rotate by 30 degrees."

Geometric transformations are complex since they alter the spatial layout of an image. Generative models, particularly those based on diffusion or transformer architecture are primarily trained to capture the content and style of data distributions instead of explicit spatial properties. This means that while they are good at tasks like modifying the appearance of objects (adding texture, changing image style, or adding objects), they lack the inherent mechanisms needed to handle geometric transformations that involve spatial manipulation instead of content modification. This is because they rely on learned correlations rather than intrinsic spatial reasoning.

Existing diffusion models primarily operate on pixel-level noise or high-level latent features but do not explicitly model geometric relationships in the latent space. They tend to entangle semantic and geometric edits. For example, a text prompt like "rotate the chair" may unintentionally modify the appearance of the chair rather than purely adjusting its orientation. Geometric transformations must preserve the overall structure and consistency of the image. This means that a rotated object should remain visually coherent and correctly positioned within the scene. Most models cannot generalize well to unseen or complex geometric transformations which limits their applicability.

1.6 Proposed Solution

Instead of directly manipulating pixels, this work focuses on learning the geometric differences, particularly for rotation, between the source and target images by leveraging latent space representations to allow generative models to implicitly learn transformations. The focus on learning the latent space representation of geometry

allows the geometric differences between the source and target images to be captured effectively. By doing so, the model learns to encode and represent transformations like rotation in a way that is both efficient and meaningful for downstream tasks.

The explicit prediction of the transformation parameter, the rotation angle, from the latent representation grants interpretability and flexibility to the model, ensuring that the geometric transformation can be explicitly applied and controlled. This approach is designed to generalize well to unseen transformations. By learning the underlying principles of geometric operations in the latent space, the model can handle geometric transformations beyond those explicitly present in the training data.

2 Background

2.1 Diffusion Models

Diffusion models employ a diffusion process to convert data into a random noise distribution and then learn to reverse this process. A sample of noise is initialized and this sample is refined iteratively towards the data distribution by performing a series of denoising steps. At each step, a neural network predicts the noise that was added at each step, thus effectively learning to reverse the diffusion process. The original image x_0 is sampled from the clean data distribution ($p(x)$) and the noisy image x_t is sampled from the Gaussian distribution ($\mathcal{N}(0, 1)$).

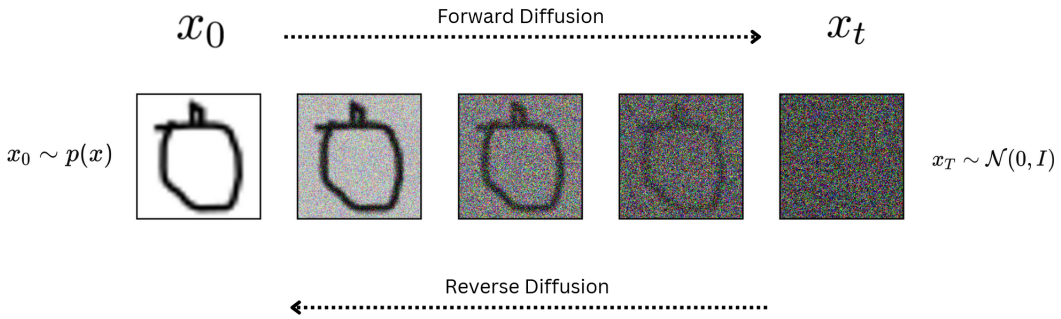

Figure 3: Visualizing the forward and reverse diffusion process

2.1.1 Forward Diffusion

This process adds noise to an original data sample incrementally until it is entirely transformed into Gaussian noise over several steps. The forward diffusion process is described by:

$$q(x_t|x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I\right) \tag{1}$$

where:

- $q(x_t|x_{t-1})$: Conditional probability of x_t given x_{t-1}.
- $\mathcal{N}$: Gaussian distribution.
- $\sqrt{1-\beta_t}$: Scaling factor for x_{t-1}.
- β_t: Noise variance at time t.
- I: Identity matrix for isotropic noise.
- x_t, x_{t-1}: States at time t and $t-1$.

Gaussian noise is added incrementally to the data, controlled by β_t, to gradually transform the original data distribution into a standard Gaussian distribution.

2.1.2 Reverse Diffusion

This is the generative phase where the model starts from pure noise and learns to denoise as it progressively reconstructs the data by estimating and subtracting the added noise at each previous step. The reverse diffusion process is described as:

$$p_\theta(x_{t-1}|x_t, c) = \mathcal{N}\left(x_{t-1}; \mu_\theta(x_t, t, c), \sigma^2 I\right) \tag{2}$$

Explanation:

- $p_\theta(x_{t-1}|x_t, c)$: Conditional probability of x_{t-1} given x_t and condition c.
- $\mathcal{N}$: Gaussian distribution.
- $\mu_\theta(x_t, t, c)$: Predicted mean parameterized by θ, dependent on x_t, time t, and condition c.
- $\sigma^2 I$: Variance term with isotropic Gaussian noise.
- x_t, x_{t-1}: States at time t and $t-1$.
- c: Additional conditioning information, such as class labels or input data.

The diffusion process allows the model to generate high-quality and coherent images, as it inherently models the complex distribution of natural images by learning the denoising trajectory.

2.2 U-Net Architecture for Denoising

U-Net [19] is a convolution network (represented in figure 4) that was originally designed for biomedical segmentation tasks but has increasingly been applied to image-to-image translation tasks, including in diffusion models.

- **Encoder-Decoder Structure**: U-Net consists of a symmetric encoder-decoder structure. The encoder compresses the input into a dense representation, and the decoder expands it back to the output resolution.
- **Skip Connections**: The model includes skip connections that transfer the feature maps from each level of the encoder to the corresponding level in the decoder. These connections help preserve the spatial hierarchies and details in the image.

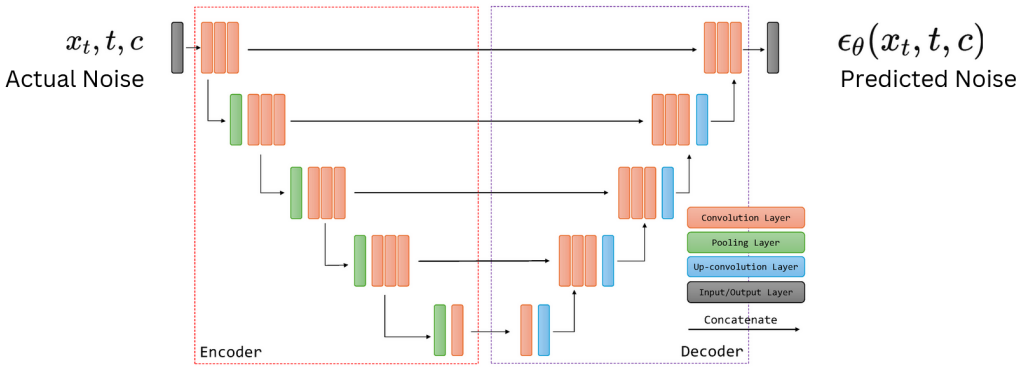

Figure 4: U-Net Architecture [20] for Denoising in Diffusion Models

In the context of diffusion models, U-Net is popularly used as the backbone architecture for denoising. The denoising process in diffusion models involves progressively refining a noisy input to reconstruct a coherent output, which aligns perfectly with U-Net's encoder-decoder structure. The encoder compresses the noisy input into a latent representation that captures the important features, while the decoder expands this representation back into the original resolution, ensuring that fine details are preserved. UNet's skip connections are ideal for transferring feature maps from the encoder layers to their corresponding decoder layers so that spatial details from the input are retained and the outputs are both visually coherent and rich in detail.

Since diffusion models often require additional inputs, such as timestep embeddings or text prompts, to guide the denoising process, U-Net accommodates these by incorporating them into the network. It leverages its hierarchical feature extraction and reconstruction capabilities to effectively model noise distributions for a smooth and accurate denoising trajectory from pure noise to a fully reconstructed image. The predicted mean for denoising in the reverse diffusion process is:

$$\mu_\theta(x_t, t, c) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \sqrt{1 - \alpha_t} \epsilon_\theta(x_t, t, c) \right) \tag{3}$$

where:

- $\mu_\theta(x_t, t, c)$: Predicted mean for the reverse diffusion step.
- x_t: The noisy input at time t.
- α_t: Time-dependent scaling factor controlling the noise level.
- $\epsilon_\theta(x_t, t, c)$: Predicted noise at time t, estimated by a U-Net model, parameterized by θ, based on x_t, t, and optional condition c.
- c: Conditioning information (text embeddings).

- $1 - \alpha_t$: Represents the noise variance added during forward diffusion.
- $\frac{1}{\sqrt{\alpha_t}}$: Normalization factor ensuring proper scaling of the denoised output.

The U-Net architecture is used to predict $\epsilon_\theta(x_t, t, c)$, the noise added to the data during the forward process. By subtracting the noise term $\sqrt{1-\alpha_t}\epsilon_\theta(x_t, t, c)$ from the input x_t, the model recovers the denoised mean $\mu_\theta(x_t, t, c)$. The process iteratively reduces noise from x_t to reconstruct the original data.

The forward and reverse diffusion processes work with the U-Net model to convert random noise to high-quality outputs. The forward process incrementally adds noise to the original data, and the reverse process denoises it step-by-step using U-Net's predictive capabilities. The integration of text prompts in the form of conditioning information allows for precise and semantically aligned transformations.

2.3 Baseline Methods

The following subsections delve into three text-guided image transformation methods used as baselines for this research.

2.3.1 FastEdit

FastEdit [21] is a method for text-guided image editing that uses semantic-aware diffusion models to understand and incorporate the meaning and context of the content it is processing. It uses LoRA (Low-Rank Adaptation) [22], which is a lightweight technique to adapt the model to specific edits while making sure that the core features of the original image remain impact. LoRA adjusts only a small subset of the model's parameters, specifically targeting the rank of weight matrices within the transformer layers, which makes it an extremely efficient technique to fine-tune. Advanced natural language processing is used to interpret the editing instructions and FastEdit manipulates the noise reduction paths of diffusion models. By controlling how noise is added and removed based on the text prompts, the model produces edited images according to the users' specifications.

2.3.2 SDEdit (Stochastic Differential Equation Editing)

SDEdit [23] is a diffusion-based guided image synthesis and editing method that implements the principles of stochastic differential equations for smooth transitions between the input image and target image based on the text prompt. Traditional diffusion models involve a forward process of gradually adding noise to an image and a reverse process of denoising it. SDEdit has an underlying diffusion model mechanism as well but modifies its approach by starting with either a partially noised image or a simple sketch, then performing a guided denoising process based on a provided

text prompt or another form of guidance. It uses a controlled way to inject noise into the image as shown in figure 6, which creates a starting point that retains some characteristics of the original or base image but includes enough stochastic variation to allow for significant transformation during the denoising process. The role of SDEs (stochastic differential equations) is to provide a mathematical framework that models the addition and removal of noise. It describes the evolution of image pixels through noise levels which grants the model better control over the editing process.

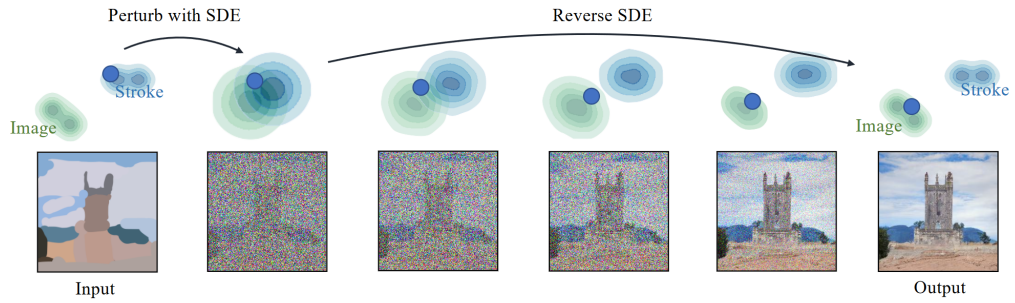

Figure 5: Visualizing SDEdit [23]: Perturbation with Gaussian Noise that is progressively removed by simulating the Reverse SDE

2.3.3 IP2P (Instruct Pix2Pix)

IP2P [12] is a diffusion-based model developed by Hugging Face as an extension to the traditional Pix2Pix [24] models for image-to-image translation. IP2P was introduced to generalize the original framework. It incorporates natural language instructions into the image-editing pipeline. The combination of GANs and language understanding models applied to fine-tuned Pix2Pix models using a dataset of (image, edit instruction, edited image) triples allows instruction-based transformations.

The IP2P method uses a denoising diffusion probabilistic model to generate images by iteratively refining noise into coherent images [12]. It uses a text encoder to convert text prompts into embeddings that condition the image generation process using cross-attention layers, along with a UNet-based neural network for the denoising process, which is conditioned on both the input image and text embeddings. Thus, IP2P leverages diffusion models to allow users to perform complex image edits using natural language prompts by guiding a diffusion model through text conditioning which allows it to generate high-quality images.

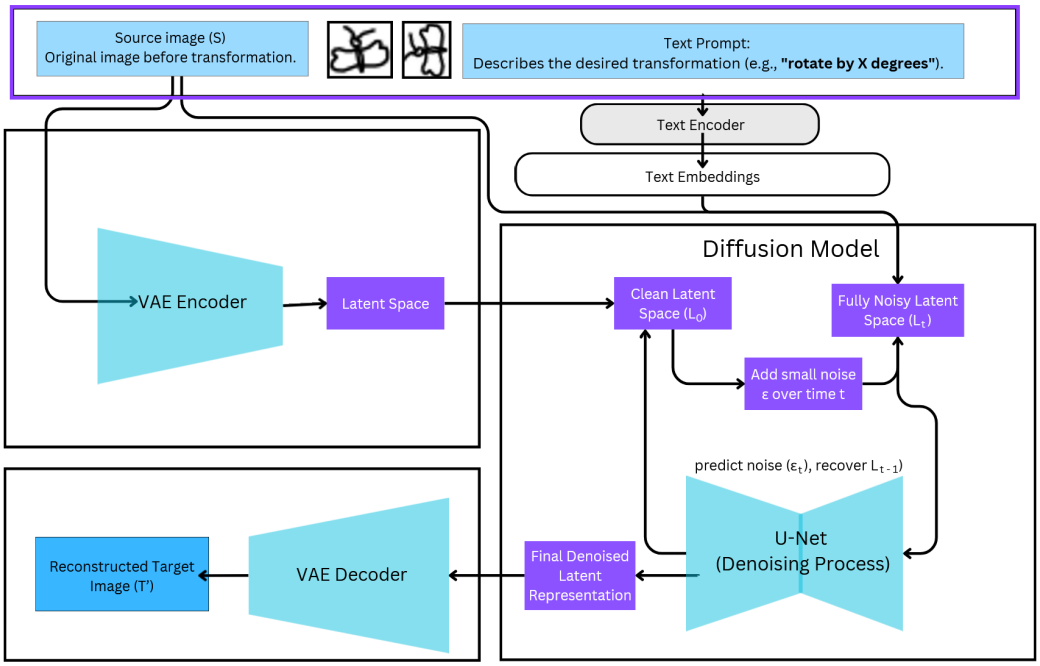

Figure 6: The Instruct Pix2Pix (IP2P) Architecture

Classifier-Free Guidance

Generative diffusion models like IP2P use the classifier-free guidance technique for tasks where the model needs to generate images that adhere to a given text description. It improves the quality and alignment of generated images by using both conditioned and unconditioned noise predictions to guide the generative process. It ensures that the generated images align with prompts while maintaining visual fidelity. The guided noise prediction, incorporating both unconditioned and conditioned passes, is defined as:

$$\epsilon_{\text{guided}}(x_t, t, c) = \epsilon_\theta(x_t, t, \emptyset) + w \cdot (\epsilon_\theta(x_t, t, c) - \epsilon_\theta(x_t, t, \emptyset)) \tag{4}$$

where:

- $\epsilon_\theta(x_t, t, \emptyset)$: Noise prediction from the unconditioned pass, capturing general image features for stability.
- $\epsilon_\theta(x_t, t, c)$: Noise prediction from the conditioned pass, guided by the text prompt c for semantic alignment.
- w: Guidance scale, a hyperparameter controlling the trade-off between semantic alignment and image quality.

The guided reverse diffusion mean is calculated as:

$$\mu_{\text{guided}}(x_t, t, c) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \sqrt{1 - \alpha_t} \epsilon_{\text{guided}}(x_t, t, c) \right) \tag{5}$$

where:

- $\mu_{\text{guided}}(x_t, t, c)$: The denoised mean, ensuring the reverse diffusion aligns with the guided noise prediction.
- x_t: The noisy input at time t.
- α_t: A time-dependent scaling factor that controls the noise level.

Classifier-Free Guidance makes sure that the generated images adhere to both semantic and geometric constraints and is a powerful technique to align text prompts with generated transformations.

3 Objective

While diffusion models have impressive capabilities for image synthesis and editing, they face limitations when it comes to learning and applying geometric transformations. Even after fine-tuning, the performance depends on the quality diversity, and extensivity of training data. If the dataset used for fine-tuning doesn't represent various rotation scenarios, the model might not generalize well. Real-world applications often require models to understand and adapt to transformations like rotation, scaling, and translation in a computationally efficient and unsupervised manner.

This thesis addresses these challenges by shifting the transformation learning to the latent space. This allows the model to learn geometric properties implicitly, without relying on an extensive amount of labeled data. Instead of passing the transformation parameter as an explicit input, we're training the model to predict and learn this parameter as part of its internal mechanism. This method is a proposed step forward in advancing the flexibility and robustness of generative models.

The primary objective of this work is to fine-tune and develop a method that forces the model to implicitly learn and apply geometric transformations. Current approaches operate in the pixel space which limits their flexibility and computational efficiency. This research aims to address these limitations by introducing a novel framework for transformations in the latent space and seeks to evaluate the proposed method through comprehensive evaluation metrics. This approach is expected to generalize beyond rotations, allowing the model to learn various transformations such as scaling, translation, and perspective shifts.

The goal is to advance generative modeling by bridging the gap between abstract latent space learning and explicit geometric understanding, with applications in domains requiring precise image manipulation and synthesis.

4 Related Work

This section explores the significant advancements and foundational methods in fine-tuning generative models, manipulating latent spaces, and applying geometric transformations. These elements are integral to this thesis, which focuses on modifying a pre-trained diffusion model, InstructPix2Pix, to implicitly learn geometric transformations.

4.1 Fine-tuning Generative Models

Fine-tuning generative models is central to this research, as it involves adapting pre-existing models to new tasks and datasets. Dhariwal and Nichol [25] explored the potentials of diffusion models for high-resolution image synthesis, achieving results competitive with those of GANs, and highlighted the importance of fine-tuning diffusion models to enhance sample quality and adapt to specific datasets. Rombach et al. [11] introduced Latent Diffusion Models (LDMs), which operate in the latent space of a pre-trained autoencoder. They showed that fine-tuning LDMs on specific tasks allows for efficient and high-quality image generation with reduced computational demands.

4.2 Manipulating Latent Spaces

A latent space is a lower-dimensional representation of the data. It captures the most salient features of the data, allowing the model to manipulate the features independently of the raw space. Latent spaces are obtained from encoding mechanisms like Variational Autoencoders (VAEs) or Convolutional Neural Networks (CNNs) [26], which compress input data into a lower-dimensional representation [27]. In VAEs, the encoding is done by mapping the input data into a probabilistic latent space defined by a mean and variance, for smoothness and continuity in the latent space. CNN-based encoders focus on extracting hierarchical features [28] through a series of convolutional layers, providing a deterministic latent representation that captures spatial and structural attributes of the data [29].

Manipulating the latent space of generative models refers to the controlled modification of generated outputs by modifying the underlying latent codes. Latent space manipulations are a powerful way to modify and generate variations of the data [30] [31]. This enables tasks such as image editing, style transfer, and applying specific attributes or transformations.

Building on the concept of vector arithmetic in embedding spaces, introduced for word representations [32], this approach was extended to image generation, showing that vector arithmetic in the latent space of GANs can achieve semantic image editing by adding or subtracting latent vectors corresponding to certain attributes

[33], implying that adding or subtracting latent vectors corresponding to certain attributes results in images with the desired modifications. Upchurch et al. (2017) [34] further explored the manipulation of visual attributes by interpolating in the latent space. They demonstrated that smooth transitions between images can be achieved, and specific attributes can be controlled by navigating the latent space in certain directions.

Manipulating latent spaces is fundamental to the proposed method, which relies on altering latent representations to implicitly learn and apply geometric transformations. The reviewed works highlight techniques for controlling image attributes through latent space manipulation, providing a foundation for developing models that can predict transformation parameters like rotation angles. By extending these concepts to diffusion models and integrating them with text conditioning, we aim to achieve precise and interpretable image transformations.

4.3 Geometric Transformations with Generative Models

The application of geometric transformations within generative models has seen significant developments, particularly with the advent of Spatial Transformer Networks (STNs) [35] and Geometry-Aware Generative Adversarial Networks (GAGAN) [36]. These methods help the model to understand and manipulate spatial data for context-aware image manipulation.

4.3.1 Spatial Transformer Networks (STNs)

Introduced by Jaderberg et al. in 2015, Spatial Transformer Networks (STNs) [35] were introduced as a novel approach to enhance the geometric invariance of convolutional neural networks without extensive data augmentation or complex preprocessing. STNs incorporate a learnable module that allows neural networks to learn how to perform spatial transformations on input data within the network itself and thus, explicitly allows for the spatial manipulation of data within the network figure 7. This module called the spatial transformer, is a differentiable attention mechanism that can be embedded into existing CNN architectures, enabling them to adapt to geometric variations in input data dynamically.

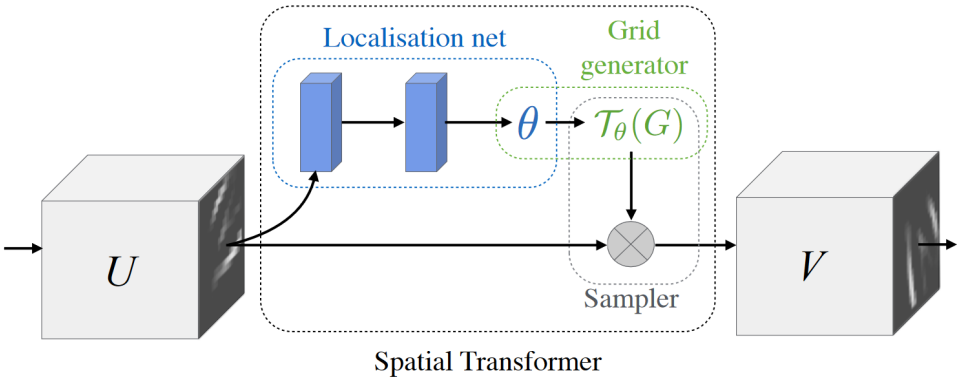

Figure 7: Spatial Transformer Network (STN) Architecture [35]

The components of STNs include:

Localization Network
This part of the STN learns the parameters of the desired transformation by analyzing the input feature map. It outputs the parameters for the spatial transformation that needs to be applied, such as scaling, rotation, or skewing.

Grid Generator
Using the transformation parameters provided by the localization network, the grid generator creates a sampling grid. This grid is a set of points from which the input image should be sampled to produce the transformed output.

Sampler
The sampler uses the grid to perform the actual sampling from the input feature map, producing the spatially transformed output. This step applies the learned transformation to the input image, enabling the model to handle variations in orientation, scale, and position.
The introduction of STNs has significantly impacted the field of computer vision, by improving the robustness of neural networks to geometric variations and providing a mechanism for models to learn spatial manipulations autonomously. The concept of learnable transformations within a neural network framework can be extended to include not only spatial transformations as learned by STNs but also transformations guided by semantic inputs.

4.3.2 Geometry-Aware Generative Adversarial Networks (GAGAN)

Developed in 2018, GAGAN ([36] introduces a novel method for incorporating geometric information directly into the generative process of GANs as shown in the figure 8. Traditional GANs often struggle with maintaining the geometric structure of objects, which can compromise the visual realism of the generated images. GAGAN

addresses these limitations by embedding geometric fidelity into the adversarial training framework.

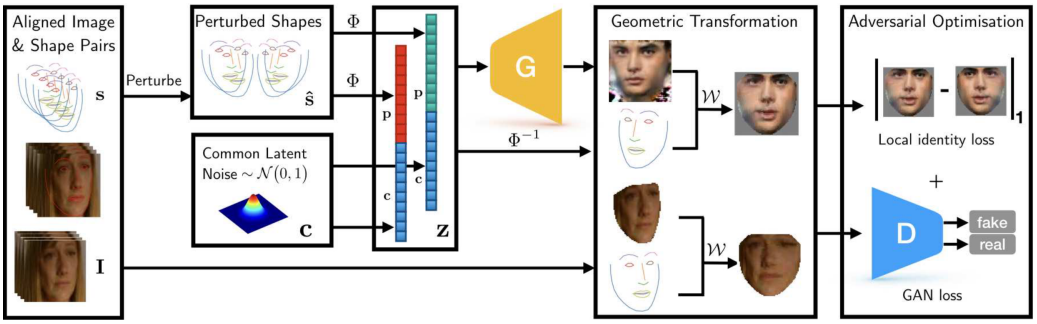

Figure 8: Overview of the Geometry-Aware GANs Model [36]

GAGAN samples latent variables from the probability space of a statistical shape model. This allows the generator to maintain the geometric integrity of generated objects by aligning generator outputs to a canonical coordinate frame through a differentiable geometric transformation. The generator uses a statistical shape model to enforce the geometry of objects, This feature is particularly beneficial for generating images with specific attributes like facial expressions, poses, and morphology. GAGAN can be seamlessly integrated into any existing GAN architecture, which enhances the morphological credibility of the generated images by leveraging prior geometric knowledge from the data distribution.

While STNs and GAGAN have advanced the field by introducing mechanisms to handle geometric transformations within neural networks, challenges remain in applying these transformations in an unsupervised and generalized manner, particularly in the context of text-guided image generation. These advancements lay the groundwork for this thesis, which aims to implicitly learn geometric transformations to guide geometric transformations in images through text-driven inputs. By integrating concepts from LDMs, STNs, and GAGAN, we aim to address the limitations of traditional models in learning and applying geometric transformations in an unsupervised manner, thereby improving results, reducing the reliance on extensive labeled datasets, and enhancing the flexibility of generative models across various domains.

5 Methodology

The following section outlines the approach taken to develop the proposed generative model for learning and applying geometric transformations implicitly through latent space manipulation. This framework integrates various components, including data preprocessing, model architecture, and training protocols for text-guided geometric transformations. The proposed methodology leverages the Google Quick, Draw! dataset to provide diverse scenarios, focusing on rotation as a proof of concept. By combining text embeddings, latent space representations, and a diffusion-based denoising pipeline, the model aims to address the limitations of current generative methods in handling spatial transformations.

5.1 Proposed Model Architecture

The foundation of the proposed model architecture for implicitly learning and applying geometric transformations through latent space manipulation is the IP2P framework. This proposed architecture as shown in the figure 9 integrates latent space learning, text guidance, and diffusion-based generative modeling to enable text-guided geometric transformations.

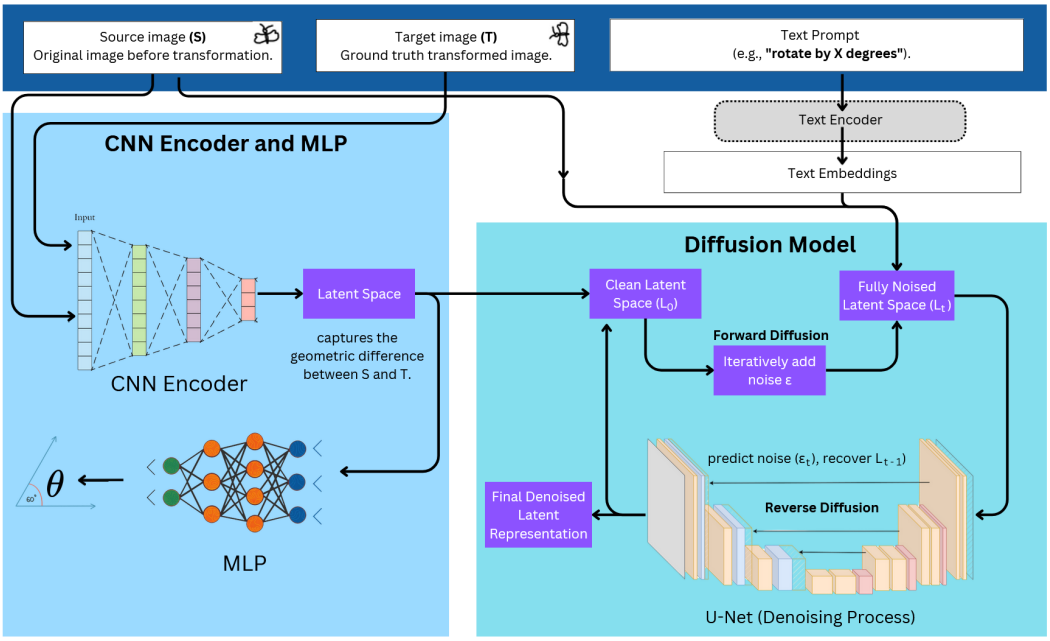

Figure 9: Proposed Model Architecture

The generation process can be divided into the following main phases:

- **Input Processing:** The system takes the source image (S) (original image before transformation) and a text prompt (e.g., "rotate by 45 degrees") as

inputs. The target image (T) (ground truth transformed image) is used during training to supervise the process.

- **Latent Space Learning:** A CNN encoder processes S and T to encode their geometric relationship into a latent representation. This latent representation captures the transformation between the source and target images.
- **Transformation Parameter Prediction:** An MLP ThetaPredictor network predicts the transformation parameter (θ_{pred}, i.e. the rotation angle) from the refined latent space.
- **Diffusion-Based Denoising:** Noise is added to the latent representation, and a diffusion model (U-Net) denoises it using text embeddings from the prompt and the downsampled source image for structural alignment. This process refines the latent space to ensure that it encodes the intended transformation.

5.1.1 Detailed Component Descriptions

CNN Encoder

The CNN Encoder learns to encode the geometric transformation between the source (S) and target (T) images. It is essential because it helps to establish a manipulatable latent space that represents the transformation between the images. Encoding the images reduces the number of parameters being updated at each step, thus increasing computational efficiency. The transformation of the input data into a latent space representation can be mathematically expressed as:

$$z = f_\theta(x) \tag{6}$$

where f_θ is the encoding function parameterized by θ, and x is the input data.

MLP ThetaPredictor

This is a specialized MLP network with layers designed to extract features and regress the transformation parameter - the rotation angle (θ_{pred}) from the latent space generated by the encoder. The final layer of the network is a tanh layer which generates an output in the range of [0,1]. The complete network architecture of MLP is shown in the figure **fig:networkLayers**. This is scaled by a factor of 180 to predict the final angle θ_{pred}. The output of this network is thus constrained to the range $[-180°, 180°]$. The transformation parameter is validated by comparing the rotated source image with the target image.

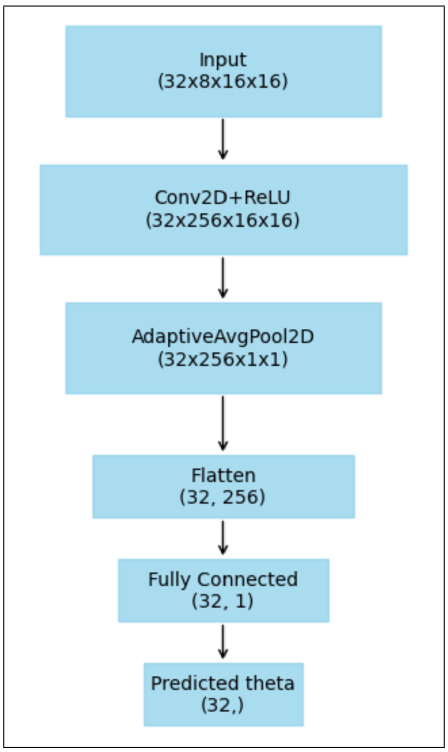

Figure 10: Network Layers of the ThetaPredictor MLP

The mean squared error (MSE) loss [37] is a commonly used loss function in regression problems, particularly in machine learning contexts where continuous outputs are predicted. For the ThetaPredictor in this work, the MSE loss evaluates the difference between the target image (T_i) and the rotated source image ($R(S_i, \theta_{\text{pred}})$) based on the predicted rotation angle (θ_{pred}). To ensure stability and prevent overfitting, a regularization term is included in the loss function. The loss function for the Encoder and MLP model is:

$$\mathcal{L}_{\text{Encoder+MLP}} = \frac{1}{n} \sum_{i=1}^{n} \|f(S_i; w) - T_i\|^2 + \lambda \cdot \text{Reg}(w)$$

where:

- $\mathcal{L}_{\text{Encoder+MLP}}$: The total loss for training the Encoder+MLP network.
- n: The number of training samples.
- S_i: Input data (e.g., features) for the i-th sample.
- T_i: Target or ground truth value for the i-th sample.
- $f(S_i; w)$: Output of the network with parameters w, given input S_i.

- $\|f(S_i; w) - T_i\|^2$: Mean squared error (MSE) between the predicted and target image.
- λ: Regularization coefficient that controls the influence of the regularization term.
- Reg(w): Regularization term applied to the network parameters w to prevent overfitting.

CLIP Text Encoder

It converts text prompts into meaningful text embeddings and provides semantic guidance to the model, allowing it to understand and act upon the instructions in the text prompt [9]. It converts the text prompts into meaningful embeddings in the form of high-dimensional vector representations of the semantic content in the text. First, the text prompt is tokenized into smaller units and each token is mapped to a corresponding embedding, a dense numerical representation that captures its semantic meaning. The individual token embeddings are combined to get a single text embedding that represents the entire prompt which represents the overall meaning of the text instruction. This embedding is incorporated into the pipeline through attention mechanisms. The text embeddings are computed using the CLIP model's text encoder as follows:

$$\mathbf{e}_{\text{text}} = \text{CLIP\_TextEncoder}(\text{Tokenizer}(\text{text})) \tag{7}$$

where Tokenizer(text) converts the input text into a sequence of tokens that are input to the CLIP text encoder. The output $\mathbf{e}_{\text{text}}$ is a high-dimensional vector representing the semantic content of the text.

Noise Scheduler

The noise scheduler controls the amount of noise being added to the latents at each timestep during training, thus simulating the diffusion process, which is essential for denoising the latent feature [38]. For the noising process, noise is incrementally added to the data. The mathematical representation of the noising process can be described by the equation:

$$x_t = \sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t}\epsilon \tag{8}$$

where x_0 is the original data, ϵ is the noise vector, and α_t is the variance of the noise at step t.

Diffusion Model

The role of this component is to refine the latent space using a denoising process to integrate the text embeddings. A noisy latent space (L_T) is generated by adding noise to the output of the encoder. Text embeddings and the downsampled source image

are passed to provide additional structural information. A U-Net architecture predicts the noise (ϵ_T) and recovers the denoised latent representation (L_0) over multiple time steps. This ensures that the latent space accurately represents the transformation described in the text prompts. The output is a clean latent representation of the geometric difference between the source and target images. The reverse diffusion can be represented as:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \alpha_t}} \epsilon_\theta(x_t, t) \right) \tag{9}$$

where ϵ_θ represents the predicted noise at step t.

The MSE loss for the Diffusion Model is given by:

$$L_{\text{Diffusion}} = E_{L_t, \epsilon} \left[\|\epsilon - \epsilon_\theta(L_t, t, \text{emb})\|^2 \right]$$

where:

$$L_t = \alpha_t L_0 + \sqrt{1 - \alpha_t}\, \epsilon$$

- L_t is the noisy latent representation at timestep t, with α_t representing the noise scheduling parameter.
- ϵ is the ground truth noise added to the latent representation L_0.
- $\epsilon_\theta(L_t, t, \text{emb})$ is the noise predicted by the U-Net-based diffusion model, conditioned on the timestep t, text embeddings, and the source image.

5.1.2 Model Workflow

Training Phase

The CNN encoder and MLP work together to learn latent representations and predict transformation parameters. The MSE loss between the rotated source image ($R(S, \theta_{pred})$) and the target image (T) is calculated and the encoder and MLP weights are frozen after training them. The U-Net is trained to denoise these latent representations using text embeddings (for geometric guidance) and the source image (for structural alignment). The MSE loss between the predicted latent (L_{T-1}) and the ground truth latent (L_0) is used to train the diffusion model to generate the denoised latent representing the geometric difference between the source and target image.

Inference Phase

The source image S and the text prompt (e.g., "rotate by 45 degrees") are provided to the trained diffusion model. The diffusion model starts from random noise and refines it to generate the latent representation guided by the text prompt and the downsampled source image. The final denoised latent representation is passed to the MLP to predict the rotation parameter (θ_{pred}) from the denoised latent.

5.2 Experimental Setup

5.2.1 About the Dataset

For training and validating the proposed approach to implicit geometric transformations, a dataset that can challenge the model to generalize across different types of geometric transformations is important. The dataset used is the Google Quick, Draw! dataset which is a collection of hand-drawn sketches collected from the Quick, Draw! online game developed by Google. In this game, users have 20 seconds to draw an object or concept that they are prompted with, and an AI tries to guess what they are drawing. It contains over 50 million drawings of animals, objects, and more, spanning more than 340 categories. The wide range of categories makes it ideal for a study focused on geometric transformations. Ten of these categories were selected to fine-tune and develop the models with a balanced representation of different shapes and complexities.

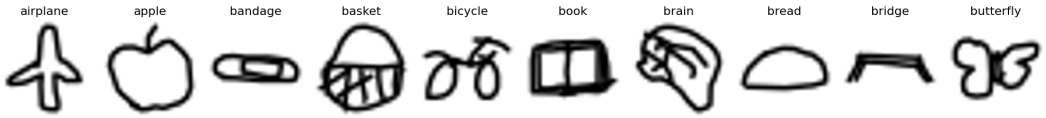

Figure 11: Samples from the Google Draw dataset

For each category, 200 distinct sketch samples were chosen as the base set. These images served as the base from which the transformed versions were generated. The rotation task is defined with a range from -180 to +180 degrees. To ensure comprehensive coverage, this range is divided into 24 bins, each representing a 15-degree interval. For each bin, five angles are randomly chosen, ensuring variability in the rotation angles applied. A rotation transformation in a 2D space for an angle θ can be represented by the following matrix:

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{10}$$

This matrix is applied to the pixel coordinates of the image to perform rotation. Each base image undergoes rotation according to the selected angles, resulting in multiple versions of the same sketch at different orientations. The distribution of angles in the dataset is uniform and there are between 3000 to 3500 samples for every angle. This method provides a robust dataset for training the model to recognize and handle various rotational transformations.

5.2.2 Data Preprocessing

The data preprocessing pipeline involves the following methods:

- **Reshaping**: The sketches, originally available as flattened arrays on Google Cloud, are reshaped back into their original 28x28 pixel grid format.
- **Scaling**: To accommodate the input requirements of the model, images are resized to a standard dimension of 128x128 pixels, providing sufficient resolution for capturing detailed transformations.
- **Normalization**: Pixel values are normalized to a range of [-1, 1] to facilitate more stable training dynamics and convergence.
- **Saving**: Finally, the processed images are saved in PNG format, preserving their quality and ensuring compatibility with the training pipeline.

For each rotated image, a corresponding text prompt is generated in the format "rotate by X degrees," where X denotes the specific rotation angle. These prompts are essential for training the model under the instruction-based framework, aligning visual changes with text commands.

Final Dataset Structure: The final dataset consists of image pairs (original and rotated) with a resolution of 128x128 each, along with the text prompt associated with the rotated image (for example - rotate by 78 degrees). Features of the dataset include:

- Original Image: The original image before transformation for reference.
- Rotated Image: The image after rotation.
- Description: A text prompt describing the transformation, e.g., "rotate by 30 degrees."

Text prompts are tokenized and converted into embeddings using the CLIP tokenizer and text encoder, using mean pooling over token embeddings to get fixed-size representations. Tokenization converts the prompts from text to a series of tokens and embeddings transform these token sequences to high-dimensional embeddings that capture semantic information.

5.2.3 Training Configuration

This section outlines the training configuration for each model used in the study. The number of training steps was decided focusing on when training should cease based on the stabilization of the loss to ensure that computational results are used optimally and prevent overfitting. Some other training parameters include:

Batch Size: A consistent batch size of 32 was maintained for all models to ensure uniformity in the amount of data processed per iteration across different training scenarios.

Learning Rate: All models were trained with a learning rate of 1×10^{-5}. This rate was chosen to facilitate gradual and stable convergence, which is crucial when models are trained until the loss stops changing significantly.

Optimizer: The Adam optimizer was used for its advantages in handling adaptive learning rates and efficiently managing sparse gradients. This optimizer is well-suited for the models in this study, particularly in scenarios where training durations are variable and dependent on loss stabilization.

6 Results and Discussion

6.1 Quantitative Evaluation Metrics

6.1.1 IS (Inception Score)

The Inception Score metric [39] evaluates the quality and diversity of generated images by using a pre-trained network, such as Inception v3 to classify images. A higher score indicates a better quality of generated images.

$$\text{IS} = \exp\left(E_{\mathbf{x}\sim p_g}\left[\text{KL}(p(y|\mathbf{x})\|p(y))\right]\right) \tag{11}$$

where:

- p_g: The distribution of generated output images.
- $p(y|x)$: The conditional class distribution given an image x, as predicted by the pre-trained model.
- $p(y)$: The marginal class distribution, computed as the average of $p(y|x)$ across all generated images.
- $\text{KL}(p(y|x)\|p(y))$: The Kullback-Leibler divergence between the conditional and marginal distributions.
- $E_{x\sim p_g}[\cdot]$: The expectation over the distribution of generated images.

6.1.2 FID (Frechet Inception Distance)

The FID score [40] compares the distribution of generated images to real images using features from a pre-trained network. Higher scores indicate better quality.

$$FID = \|\mu_1 - \mu_2\|^2 + \text{Tr}(\Sigma_1 + \Sigma_2 - 2(\Sigma_1\Sigma_2)^{\frac{1}{2}}) \tag{12}$$

where:

- μ_1 and μ_2 are the mean feature vectors of the real and generated images, respectively.
- Σ_1 and Σ_2 are the covariance matrices of the real and generated images, respectively.
- Tr denotes the trace of a matrix (sum of its diagonal elements).

6.1.3 SSIM (Structural Similarity Index)

The SSIM metric [41] evaluates structural similarity, focusing on luminance, contrast, and structure.

$$\text{SSIM}(I, K) = \frac{(2\mu_I\mu_K + C_1)(2\sigma_{IK} + C_2)}{(\mu_I^2 + \mu_K^2 + C_1)(\sigma_I^2 + \sigma_K^2 + C_2)} \tag{13}$$

where:

- μ_I and μ_K are the mean pixel values of images I and K, respectively.
- σ_I^2 and σ_K^2 are the variances of images I and K, respectively.
- σ_{IK} is the covariance between I and K.
- C_1 and C_2 are constants to prevent division by zero.

6.1.4 ThetaPredictor Metrics

The model's performance in predicting transformation parameters (rotation angles) was evaluated using Mean Squared Error (MSE) (the average squared difference between predicted and true rotated images,) and Mean Absolute Error (MAE) (the average absolute difference between predicted and true rotation angles, providing a direct measure of prediction accuracy.). These metrics help in understanding how well the model aligns predicted rotations with ground truth values. For the encoder and MLP phase, the following metrics were computed:

- **Test MSE:** 0.64, indicating low squared error in predicted rotations.
- **Test MAE (theta_loss):** 4.33 degrees, showcasing high accuracy in predicting rotation angles.

For the diffusion model, a 95% confidence interval was computed for the predicted rotation angles to quantify the uncertainty in the predictions. The margin of error (MOE) was calculated as follows:

$$\text{MOE} = t_{0.025,n-1} \cdot \frac{\sigma}{\sqrt{n}} = \pm 7.11 \tag{14}$$

This leads to a confidence interval:

$$\text{CI} = \left[\bar{\theta} - 7.11, \bar{\theta} + 7.11\right] \tag{15}$$

This interval quantifies the range within which 95% of the predicted values are expected to fall, given the inherent stochasticity of the reverse diffusion process.

6.2 Comparative Analysis

The table 1 represents loss metrics discussed in the previous section for the baseline methods (FastEdit, SDEdit, and IP2P) and the proposed methods, calculated for the test images.

Method	**FID Score ↓**	**IS ↑**	**SSIM ↑**
FastEdit	13.857	1.0740	0.2854
SDEdit	10.644	1.0713	0.5263
IP2P	11.794	2.0689	0.6570
Proposed Model	4.857	3.5278	0.8860

Table 1: Quantitative Loss Metrics Comparison

FastEdit performs the worst across all metrics, with the highest FID score (13.857), indicating a significant deviation from the real image distribution. Its low Inception Score (1.0740) highlights a lack of diversity and poor image sharpness, while the SSIM (0.2854) shows severe structural inconsistencies, showing that this method struggles to generate realistic or structurally accurate images.

SDEdit shows moderate improvement over FastEdit, with a reduced FID score (10.644) and slightly better diversity and clarity, as indicated by its marginally higher Inception Score (1.0713). The SSIM (0.5263) shows better structural alignment with the ground truth but remains far from ideal, suggesting this method has potential but lacks robustness.

IP2P performs moderately well, with an improved Inception Score (2.0689), suggesting better diversity and sharpness than FastEdit and SDEdit. Its FID score (11.794) is still relatively high, indicating distribution misalignment, but the SSIM (0.6570) reflects decent structural similarity. While better overall, IP2P struggles with realism.

The proposed model significantly outperforms all other methods, achieving the lowest FID score (4.857), indicating excellent alignment with the real image distribution. Its Inception Score (3.5278) highlights high diversity and image sharpness, while the SSIM (0.8860) demonstrates near-perfect structural similarity to the ground truth, making it the most robust and reliable method in this comparison.

6.3 Qualitative Evaluation

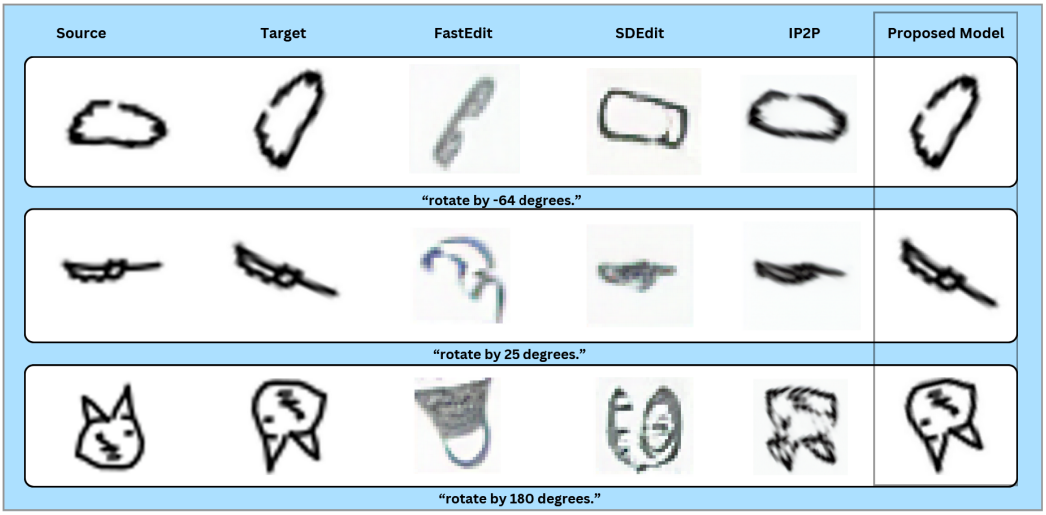

Figure 12: Comparison Results for the Fine-Tuned Baseline Models and Proposed Model)

The tuned FastEdit method results shown in the figure 12, show the use of other classes in results (e.g., shapes not related to the category, thus misclassifying objects or pattern classes in results. Incorporating features from other classes leads to severe misclassification, where results are distorted and irrelevant shapes come into play. Patterns from unrelated classes make the outputs noisy and incoherent, further reducing realism.

The fine-tuned SDEdit model preserves the structure of the source image better than the FastEdit method. However, it introduces a wide variety of noise and artifacts and struggles with preserving the finer details of the images.

The fine-tuned IP2P model shows better rotational alignment than both the FastEdit and SDEdit methods. Structural consistency is improved, as the shapes retain clearer edges. However, texture inconsistencies are present in many generated images and complex shapes tend to lose fine details.

The proposed model shows significantly better geometric alignment than all other methods. The model effectively captures rotational transformations without breaking the topography and can handle complex shapes with ease. The alignment between output images and ground truth rotations is usually close.

6.3.1 Summary of Results

Table 2 provides a comparative analysis of the strengths, weaknesses, and overall performance of different methods:

Method	**Strengths**	**Weaknesses**	**Overall Performance**
FastEdit	Basic alignments preserved.	Very low generalization to categories and noisy outputs.	Struggles with both realism and diversity.
SDEdit	Improved structural consistency for simpler shapes.	Limited diversity, struggles with details.	Moderate improvement but lacks robustness and does not learn rotations.
IP2P	Better alignment and diversity.	Noise, texture inconsistencies.	Better generation but breaks topography in most cases.
Proposed Model	Outputs are aligned with the text prompt.	Predicted rotation is off in some cases	Comparable performance overall.

Table 2: Comparative Analysis of Methods

6.4 Discussion

6.4.1 Why This Architecture is Effective

This section discusses the main factors that contribute to the effectiveness of this architecture.

Implicit Learning of Transformations
Instead of relying on explicitly labeled rotation angles corresponding to each category, the model learns to associate latent space differences with specific geometric transformations. It learns to predict the rotation angle based on the patterns it discovers in the latent space which allows it to generalize to unseen images and categories. This reduces the dependency on extensively labeled datasets, which are costly and time-consuming to produce. The latent space encodes high-level abstractions of the input data, which allows the model to generalize the concept of rotation instead of specific pixel manipulations.

Flexibility for Transformations
While this work focuses on rotations, the architecture can generalize to other geometric transformations. For example:

- Scaling: The architecture can be adapted to learn a scaling factor (e.g., "scale by a factor of 1.5"). Example Application - To augment medical datasets where images of organs must be resized to match anatomical proportions.
- Shearing: Shearing transformations (e.g., tilting an image by a shear factor) could be encoded similarly. For instance - applying a shear to sketches could be useful for perspective changes, to view an object from a different angle.
- Combination Transformations: By working in latent space, the model can predict combined transformations like rotating and scaling simultaneously, which would otherwise require separate processing.

Interpretable Outputs and Text Guidance
The model uses a clear and interpretable parameter that represents the degree of rotation applied to the image. This helps in understanding the model's behavior and verifying that it performs as intended. The explicit prediction is valuable in applications requiring precision. By explicitly predicting rotation angles, the model ensures that users can verify its decisions. For example, in an augmented reality setup, if a virtual object fails to align correctly with a real-world reference frame, the user can inspect the predicted angle (θ_{pred}) and adjust it if needed.

The integration of text prompts also ensures interpretability and user-friendly control.

6.4.2 Limitations of the Proposed Model

Dependence on Parametric Transformations

The model is designed to predict and apply geometric transformations that can be defined by explicit parameters (e.g., rotation angle, scaling factor, etc). However, more complex, non-parametric transformations, such as elastic deformations may require significant architectural adjustments.

Extending the latent space representation to incorporate localized, non-parametric transformations or integrating a spatial transformer module could enable the model to handle elastic deformations.

Limited to Single Transformation Type

The current implementation performs well for single transformations involving only one parameter (like the rotation angle). For applications where simultaneous transformations (e.g., rotation + scaling) are required, an extended version of the model would need to predict multiple parameters. This will require significant architectural modifications and may introduce additional computational costs.

A multi-task learning framework where the model predicts multiple parameters (e.g., θ for rotation, t_x and t_y for translation, and s for scaling) could address this limitation.

Lack of Localized Control

The architecture, in its current form, works for global transformations, where all parts of the image are affected uniformly. This is well-suited for tasks such as alignment or resizing that require consistent, global changes. However, it struggles when transformations need to be applied to specific regions while keeping the rest of the image the same.

Training Data Requirements

Although the model reduces the need for explicit labels for each category to learn transformations, it still requires a substantial amount of data representing the transformations to learn effectively. If the training data does not cover a wide range of rotation angles and contexts, the model may fail to generalize to unseen cases.

For example - if the training dataset lacks rotations for asymmetric objects (e.g., a chair viewed at extreme angles), the model may struggle to predict accurate transformations for these objects during inference.

7 Future Work

7.1 Expanding beyond Rotations

While the primary focus of this research has been learning rotations, the ultimate goal is to create a generative model that can handle a broader array of geometric transformations. Expanding the model's capabilities to include scaling, translation, shearing, and perspective transformations can greatly enhance its utility across various applications.

The proposed architecture can be made more versatile by enhancing the model to handle a wider range of geometric transformations. These complex image manipulations find applications in a variety of domains including advanced graphic design, augmented reality, and medical imaging.

7.2 Enabling Localized Tranformations

The model's capability can be extended to handle region-specific transformations for localized edits while preserving the rest of the image. This capability can be integrated by predicting complex parameter spaces, i.e., grids of transformation parameters for localized control. This is useful for adjusting the relevant elements of the image and transforming specific regions without affecting the surrounding image.

7.3 Real-World Applications

7.3.1 Medical Imaging

In medical imaging, accurate geometric transformations are essential for image registration tasks including aligning or augmenting images [42]. Different modalities (such as MRI, CT scans, and X-rays) or scans taken at different times must be aligned accurately. A geometrically aware diffusion model will enhance this process by automatically adjusting and aligning images based on their internal geometric structures. This improves the accuracy of medical diagnoses by providing clearer, more consistent images [43]. For example - in tumor tracking across sequential images, the model will ensure that each image is perfectly aligned over time for accurate analysis. Another application is the transformation of pre-operative scans to predict post-operative outcomes or simulate surgical adjustments.

7.3.2 Robotics and Autonomous Systems

In robotics, understanding and predicting geometric transformations is important for object recognition and navigation as generative models can enhance robots' spatial awareness. This is useful to recognize and manipulate objects that are oriented in

various positions. Generative models can help predict transformations (e.g., rotation, scaling) needed to align an object with the robot – robots can also learn to align their tools or sensors with target objects by predicting geometric transformations [44] [45]. This would enhance the efficiency of the robot and its ability to perform complex tasks autonomously.

7.3.3 Augmented and Virtual Reality

In AR and VR applications, object manipulation significantly can enhance the user experience by allowing dynamic interaction with digital content [46]. Generative models can adjust virtual objects to align with the user's perspective or environment. For example - in an AR application, the model could adjust the appearance of virtual objects based on the virtual space's dimensions and perspective [47]. Similarly, in VR, these models could help in creating more immersive worlds where users can manipulate objects in real-time, thus enhancing the engagement of virtual experiences.

7.4 User Interaction and Control

To improve the user experience, and make the model more adaptable - graphical user interfaces can be developed that allow users to input custom transformation parameters or adjust predictions with real-time previews. Feedback loops can be integrated where users can provide corrections to allow the model to learn from user inputs and improve over time. Users could also customize aspects of the transformation process, such as specifying areas of interest or constraints on the transformations.

Together, these features would make the model more intuitive and adaptable, and allow for adoption in fields like healthcare, design, and automation.

8 Conclusion

The research presented in this thesis contributes to the advancement of generative modeling by introducing an architecture that enables implicit learning of geometric transformations through latent space manipulation and improves the generalization of the model as it explicitly learns to associate latent differences with interpretable transformations. The explicit prediction of transformation parameters, such as the rotation angle θ, makes the model more versatile.

The proposed architecture represents a step toward creating generative models that are both powerful and interpretable. By addressing current limitations and exploring the avenues outlined for future work, the model's impact can be further amplified, contributing to advancements in fields ranging from computer vision to human-computer interaction.

In conclusion, this research lays the groundwork for developing generative models capable of learning and applying a wide array of geometric transformations. It highlights the importance of latent space manipulation and explicit parameter prediction in achieving these goals, thus offering important insights for future developments in the field.

9 Bibliography

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial nets," in *Advances in neural information processing systems*, vol. 27, NeurIPS, 2014, pp. 2672–2680.

[2] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *arXiv preprint arXiv:1508.06576*, 2015.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[4] D. P. Kingma, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[5] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*, PMLR, 2015, pp. 2256–2265.

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

[7] T. Zhang, H.-Y. Tseng, L. Jiang, W. Yang, H. Lee, and I. Essa, "Text as neural operator: Image manipulation by text instruction," *arXiv preprint arXiv:2008.04556*, 2020.

[8] A. Ramesh, M. Pavlov, G. Goh, *et al.*, "Zero-shot text-to-image generation," *arXiv preprint arXiv:2102.12092*, 2021.

[9] A. Radford, J. W. Kim, C. Hallacy, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, PMLR, 2021, pp. 8748–8763.

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

[11] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[12] T. Brooks, A. Holynski, and A. A. Efros, "Instructpix2pix: Learning to follow image editing instructions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 18 392–18 402.

[13] R. Wang, Y. Zhang, Y. Zhang, *et al.*, "Diffusion models are geometry critics: Single image 3d editing using pre-trained diffusion priors," *arXiv preprint arXiv:2403.11503*, 2023.

[14] B. U. IVL Lab, "Geodiffuser: Geometry-based image editing with diffusion models," *IVL Research*, 2023, `https://ivl.cs.brown.edu/research/geodiffuser.html`.

[15] LearnOpenCV Team, "Image translation and rotation using opencv," *LearnOpenCV*, 2021. [Online]. Available: `https://learnopencv.com/image-rotation-and-translation-using-opencv/`.

[16] GeeksforGeeks, "Image transformations using opencv in python," *GeeksforGeeks*, 2022. [Online]. Available: `https://www.geeksforgeeks.org/image-transformations-using-opencv-in-python/`.

[17] S. M. LaValle, "The geometry of virtual worlds," in *Virtual Reality*. Cambridge University Press, 2019, pp. 63–102. [Online]. Available: `https://msl.cs.uiuc.edu/vr/vrch3.pdf`.

[18] Mathcracker, "Virtual geometry: Advanced mathematical frameworks in augmented and virtual reality engineering," *Mathcracker*, 2023. [Online]. Available: `https://mathcracker.com/article/virtual-geometry-advanced-mathematical-frameworks-augmented-virtual-reality-engineering`.

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, Springer, 2015, pp. 234–241.

[20] I. Dimitrovski, V. Spasev, S. Loshkovska, and I. Kitanovski, "U-net ensemble for enhanced semantic segmentation in remote sensing imagery," *Remote Sensing*, vol. 16, no. 12, p. 2077, 2024.

[21] Z. Chen, Z. Zhao, Y. Luo, and Z. Huang, "Fastedit: Fast text-guided single-image editing via semantic-aware diffusion fine-tuning," *arXiv preprint arXiv:2408.03355*, 2024.

[22] E. J. Hu, Y. Shen, P. Wallis, *et al.*, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[23] C. Meng, Y. He, Y. Song, *et al.*, "Sdedit: Guided image synthesis and editing with stochastic differential equations," *arXiv preprint arXiv:2108.01073*, 2021.

[24] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[25] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in neural information processing systems*, vol. 34, pp. 8780–8794, 2021.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25, 2012, pp. 1097–1105.

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[28] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *International Conference on Artificial Neural Networks*, Springer, 2011, pp. 52–59.

[29] M.-M. Cheng, P.-T. Jiang, L.-H. Han, L. Wang, and P. Torr, "Deeply explain cnn via hierarchical decomposition," *International Journal of Computer Vision*, vol. 131, pp. 1091–1105, 2023.

[30] X. Li, C. Lin, R. Li, C. Wang, and F. Guerin, "Latent space factorisation and manipulation via matrix subspace projection," *arXiv preprint arXiv:1907.12385*, 2019.

[31] L. Tronchin, M. H. Vu, P. Soda, and T. Löfstedt, "Latentaugment: Data augmentation via guided manipulation of gan's latent space," *arXiv preprint arXiv:2307.11375*, 2023.

[32] T. Mikolov, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, vol. 3781, 2013.

[33] T. White, "Sampling generative networks," *arXiv preprint arXiv:1609.04468*, 2016.

[34] P. Upchurch, J. Gardner, G. Pleiss, *et al.*, "Deep feature interpolation for image content changes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7064–7073.

[35] M. Jaderberg, K. Simonyan, A. Zisserman, *et al.*, "Spatial transformer networks," *Advances in neural information processing systems*, vol. 28, 2015.

[36] J. Kossaifi, L. Tran, Y. Panagakis, and M. Pantic, "Gagan: Geometry-aware generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 878–887.

[37] S. K. Singh *et al.*, "Analysis of loss functions for image reconstruction using convolutional autoencoders," in *International Conference on Pattern Recognition and Machine Intelligence*, Springer, 2022.

[38] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 6840–6851.

[39] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, 2016.

[40] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[41] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[42] S. Brown, Y. Zhang, and M. Chen, "A review of medical image registration for different modalities," *MDPI Bioengineering*, vol. 11, no. 8, p. 786, 2022. DOI: 10.3390/bioengineering11080786. [Online]. Available: https://www.mdpi.com/2306-5354/11/8/786.

[43] J.-H. Kim, S.-Y. Park, and J.-M. Lee, "Diffusemorph: Unsupervised deformable image registration using diffusion model," *arXiv preprint arXiv:2112.05149*, 2021. [Online]. Available: https://arxiv.org/abs/2112.05149.

[44] A. Mumuni and F. Mumuni, "Cnn architectures for geometric transformation-invariant feature representation in computer vision: A review," *SN Computer Science*, vol. 2, no. 4, p. 340, 2021. DOI: 10.1007/s42979-021-00735-0. [Online]. Available: https://link.springer.com/article/10.1007/s42979-021-00735-0.

[45] S. Yuan, H. Huang, Y. Hao, C. Wen, A. Tzes, and Y. Fang, "Gamap: Zero-shot object goal navigation with multi-scale geometric-affordance guidance," *arXiv preprint arXiv:2410.23978*, 2024. [Online]. Available: https://arxiv.org/abs/2410.23978.

[46] Y. Xing, Q. Liu, J. Wang, and D. Gomez-Zara, "Smore: Enhancing object manipulation and organization in mixed reality spaces with llms and generative ai," *arXiv preprint arXiv:2411.11752*, 2024. [Online]. Available: https://arxiv.org/abs/2411.11752.

[47] W. Zhang, M. Li, X. Chen, and H. Wang, "Vrcopilot: Authoring 3d layouts with generative ai models in vr," *arXiv preprint arXiv:2408.09382*, 2024. [Online]. Available: https://arxiv.org/abs/2408.09382.