Modality Translation through Conditional Encoder-Decoder

Hyunsoo Lee Yoonsang Lee Maria Pak Jinri Kim
Seoul National University
{philip21, lysianthus, pakmasha99, ruth9811}@snu.ac.kr

Abstract

With the recent rise of multi-modal learning, many of the developed models are task-specific, and, as a consequence, lack generalizability when applied to other types of downstream tasks. One of the representative models that overcomes this issue of generalizability is CLIP, which attacks downstream tasks using cosine similarity metric. However, CLIP has shown relatively low cosine similarity between text and image vector representations. For this reason, we aim to develop a new approach that more accurately maps the hyperplanes of text and image embeddings, and thus, achieves a high-quality text-image modality translation. To this end, we propose a new conditional encoder-decoder model that maps a latent space of one modality given another modality as a condition. We observe that our model is a general method that can be used with various latent encoders and decoders, which are not limited to multi-modal models. Experiments show that conditional encoder-decoder achieves comparable results with the previous state-of-the-art on several downstream tasks.

1. Introduction

Multi-modal learning has recently gained significant attention as a promising approach to solving various downstream tasks effectively. One of the representative models, CLIP [26], aligns image and text embeddings in the same hyperspace, and has been widely used in tasks such as text-to-image generation and image captioning. However, the cosine similarity of image and text embeddings extracted by CLIP is found to be relatively low, hovering around 0.3. Extensive studies have been conducted to enhance the cosine similarity and evaluate the performance in various downstream tasks. For instance, unsupervised T2I generation models have exploited CLIP and simply add Gaussian noise [35] to enhance the mapping within the same hyperspace as CLIP, surpassing its performance.

To overcome these challenges, we propose a novel approach that can effectively map the hyperplanes of two encoders, enabling robust solutions for a wider range of downstream tasks. Our proposed model is not dependent on a particular latent encoder-decoder model and can be applied to various architectures. We extract latent vectors of each modality using various latent encoders. For text, we use BERT [3] and the text encoder of CLIP, while for images, we utilize DINOv2 [22] and the image encoder of CLIP. Since we employ pretrained encoders, no training is required in this process. Next, we define the bidirectional translation between latent vectors of two modalities by training conditional encoder-decoder model corresponding to the conditional DDIM using a modified version of the latent DDIM proposed in the Diffusion Autoencoders [25]. We define the conditional DDIM using the latent vector of one modality as the condition to predict the latent vector of another modality.

Furthermore, we evaluate our proposed model on various downstream tasks, including image generation, image retrieval, image captioning, and image classification, and achieve comparable results. These evaluations demonstrate that our model effectively maps the hyperplanes of both encoders.

Our contributions can be summarized as follows:

- We propose a general method that does not rely on a specific latent encoder-decoder architecture.
- We propose an algorithm that aims to effectively map the hyperplanes of both encoders, facilitating seamless integration between modalities.
- We conduct extensive experiments across diverse downstream tasks to validate the performance of our method, yielding comparable results.

The remaining sections of our paper are structured as follows. In Section 2, we provide an overview of the related work on multi-modal feature representation, Diffusion Probabilistic Models, and various downstream tasks in multi-modal learning. In Section 3, we present the main idea behind our approach. The experiment and implementation details are discussed in Section 4. We present the experimental results in Section 5, and finally, in Section 6, we conclude this paper.

Model	dim
CLIP ViT-B/32 [26]	512
CLIP ViT-L/14 [26]	768
CLIP RNx50 [26]	640
BERT [3]	768
DINOv2 [22]	1024
DALL-E-2 [27]	768
CapDec [21]	640
ClipCap [20]	640

Table 1. Embedding dimensions among different models.

2. Related Work

2.1. Multi-modal Feature Representation

Transforming data into latent vectors is a crucial step in handling multi-modal data such as images, text, and audio. Two types of embedding models, namely encoders and decoders, play key roles in this process. Encoder encodes the data into an embedding vector, while decoder decodes the embedding vector back into the corresponding data. Numerous pretrained models have been developed using various training methods. The most widely recognized model for image-text multi-modal embedding is CLIP [26] and Flamingo [1], which is trained using contrastive learning with image-text pairs. Moreover, LAFITE [35] and CapDec [21] have incorporated Gaussian noise into CLIP embeddings. Pix2pix-zero [24] has improved the accuracy of text embedding by leveraging the captioning model of BLIP [15], while DALLE-2 [27] has been trained with an explicit prior model. Additionally, unimodal embedding models can also be selected as latent embedding models. Examples of these include StyleGAN [13] and VQ-GAN [6] for images, BERT [3] and SimCSE [7] for text. In our research, we adopt the CLIP model as the baseline embedding model, along with DALLE-2 as the image decoder, BERT as the text encoder, and CapDec as the text decoder. The embedding dimensions of these models used in our paper are presented in Table 1.

2.2. Diffusion Probabilistic Models

Conditional Denoising Diffusion Implicit Models. To generate a data sample, modeling data distribution is a very significant task. In recent years, diffusion-based models are widely used to model the manifold of dataset. Especially, Denoising Diffusion Implicit Model (DDIM) [32] enables more accurate modeling compared to previous generative models. First step is called *forward process*, which perturbates data points $\mathbf{x}_0$ by adding gaussian noise for T steps:

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\epsilon \tag{1}$$

where α_t is defined using variance of each forward process and $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a gaussian noise. In second step, which is called *denoising step*, network $\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c})$ is trained to fit the noise ϵ, with training objective introduced in [10]:

$$\mathbb{E}_{t,\mathbf{x}_0,\epsilon}\left[\|\epsilon - \epsilon_\theta\left(\mathbf{x}_t, t, \mathbf{c}\right)\|^2\right] \tag{2}$$

At inference, $\mathbf{x}_{t-1}$ is sampled from $\mathbf{x}_t$ using deterministic process:

$$\begin{aligned}\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}}\left(\frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c})}{\sqrt{\alpha_t}}\right) \\ + \sqrt{1-\alpha_t}\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c})\end{aligned} \tag{3}$$

Classifier-Free Guidance. To improve the sampling performance of diffusion models, Classifier-Free Guidance (CFG) [11] can be applied to diffusion models. For training, a given condition $\mathbf{c}$ is substituted to an unconditional setting ϕ with a constant probability p, and network is optimized using Eq. (2). For inference, predicted noise $\tilde{\epsilon}$ is sampled using positive weight on conditional noise sample and negative weight on unconditional noise sample:

$$\tilde{\epsilon} = (1+\omega)\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c}) - \omega\epsilon_\theta(\mathbf{x}_t, t, \phi), \tag{4}$$

where scalar ω is a guidance scale. Then, $\mathbf{x}_{t-1}$ is sampled using Eq. (3) with $\tilde{\epsilon}$ instead of $\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c})$.

2.3. Downstream Tasks

In recent years, there has been a significant increase in the number of studies exploring the use of multi-modal learning to effectively solve various downstream tasks, including text-to-image generation [27, 34, 35], image captioning [15, 20, 21], image classification [26] and retrieval [19, 34].

Text-to-Image Generation. Recent advances in text-to-image generation have shown great potential for generating high-quality images from textual description. CLIP-based methods have emerged as a promising approach, with models such as LAFITE [35] and CLIP-GEN [34] leveraging CLIP's properties to train text-to-image generation models in language-free setting. Stable Diffusion [28] also gets CLIP text embedding and uses attention mechanism to generate high-resolution images. On the other hand, DALL-E-2 [27] proposes a two-stage model that leverages CLIP representations for image generation. Our model, similar to the previously mentioned model, utilizes text embeddings to effectively address the image generation task. These models highlight the advantages of using pre-trained text-image multi-modal models to guide image generation.

Image Retrieval. Image retrieval is a task of finding images in a large-scale dataset that are most relevant to the text query given by user. Multi-modal models can be applied to this task. For example, CLIP [26] can be applied to retrieval tasks and shows good performance. CIRPLANT [19] uses

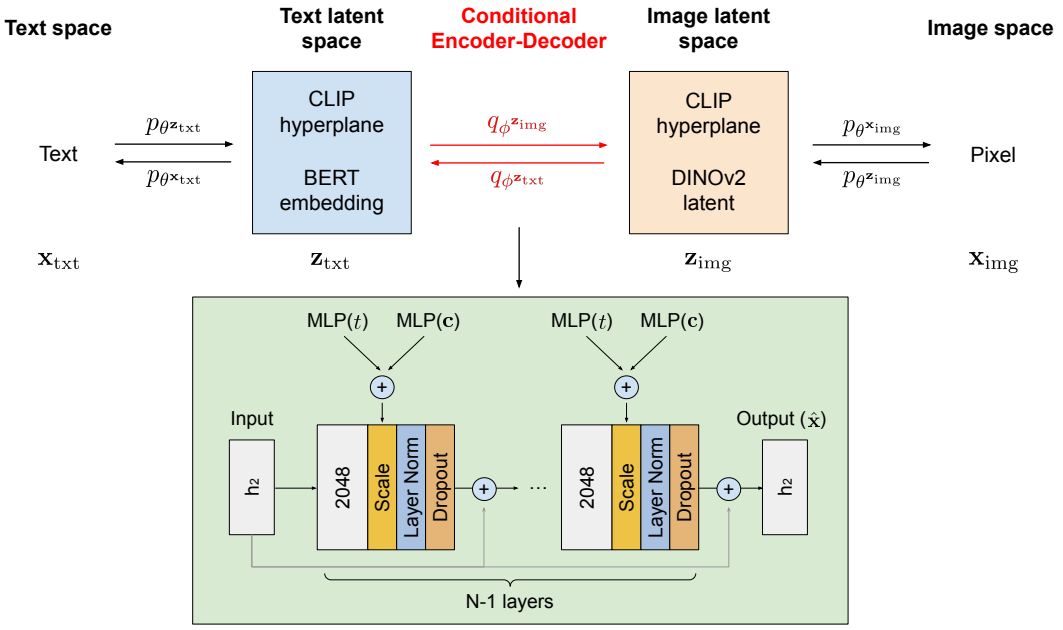


Figure 1. High-level overview of the model structure. Parts of the model that were subject to training are indicated in red.

metric learning to leverage a transformer-based model architecture, which modifies the image embedding extracted from pretrained image-text model to obtain more relevant representation conditioned on the text input.

Image Captioning. Image captioning can be achieved by leveraging two modalities (text and image) to generate captions that describe the visual content in an image. CapDec [21] is a model for image captioning that aims to train CLIP [26] solely on text samples by introducing zero-mean Gaussian noise into the text embeddings prior to decoding. Furthermore, ClipCap [20] utilizes the CLIP framework and a pre-trained language model GPT-2 to achieve a comprehensive understanding of both visual and textual data. BLIP [15] presents a novel approach to solve the image captioning task by effectively leveraging noisy web data using bootstrapping for captions.

Image Classification. Image classification is a task that involves categorizing objects in an image into different classes. CNN-based classification, such as ResNet [8], VGGNet [31], and Inception [33], has been a commonly adopted approach for this task. However, with the advent of Vision Transformer (ViT) [5], it has been shown that a pure transformer applied directly to sequences of image patches can perform well on image classification tasks. Moreover, the CLIP [26] model, which extends the transformer-based architecture into multi-modalities, has shown effectiveness in image classification, particularly in zero-shot image classification.

3. Method

3.1. Conditional Encoder-Decoder

In this work, we propose a model for the latent space prediction across different modalities. High-level structure of our model is presented in Fig. 1. Given text and image data, latent encoders $p_{\theta^{\mathbf{x}_{\text{txt}}}}$ and $p_{\theta^{\mathbf{x}_{\text{img}}}}$ extract the text and image embeddings $\mathbf{z}_{\text{txt}}$ and $\mathbf{z}_{\text{img}}$, respectively. Then a conditional encoder-decoder is used to perform a bidirectional transformation between two latent vectors. Finally, transformed embeddings are converted to the image and text modalities via latent decoders $p_{\theta^{\mathbf{x}_{\text{img}}}}$ and $p_{\theta^{\mathbf{x}_{\text{txt}}}}$.

The goal of our model, referred as conditional encoder-decoder, is to predict a latent vector of a particular modality (*i.e.* target vector), using a latent vector of another modality as a condition (*i.e.* condition vector). For instance, the conditional encoder, $q_{\phi^{\mathbf{z}_{\text{img}}}}(\mathbf{z}_{\text{img}}|\mathbf{z}_{\text{txt}})$, aims to predict an image embedding given a text embedding, while the conditional decoder, $q_{\phi^{\mathbf{z}_{\text{txt}}}}(\mathbf{z}_{\text{txt}}|\mathbf{z}_{\text{img}})$, operates in the opposite direction.

For the architecture of our model, we use a modified version of the latent DDIM proposed in [25], which has the backbone of stacked MLPs with skip connections as they were found to be well-performing and sufficiently fast. Unlike the latent DDIM, our modified model is conditioned on both time and condition vector to perform a transformation between two different modalities. We use residual connections for layers which injects condition vector to DDIM. Also, we apply classifier-free guidance [11] to improve the performance of diffusion model's sampling proce-

dure. Condition and target vectors' dimensions, which are denoted as h_1 and h_2, are changed according to the applied model as shown in Table 1.

3.2. Sampling Strategy

We denote $\mathbf{z}$ as a target vector, $\hat{\mathbf{z}}$ as a prediction of $\mathbf{z}$, and $\mathbf{c}$ as a condition vector. Then, we propose a novel method for sampling $\hat{\mathbf{z}}$, estimated target vector. Using our model, we derive the sample $\hat{\mathbf{z}}$ conditioned on $\mathbf{c}$:

$$\hat{\mathbf{z}} = \mathbf{c}\mathbf{M} + \gamma\mathbf{F}(\mathbf{c}), \tag{5}$$

where $\mathbf{M} \in \mathbb{R}^{h_1 \times h_2}$ is an additional parameter called *mapping matrix*, $\gamma \in \mathbb{R}$ is a hyperparameter, and $\mathbf{F}(\mathbf{c}) \in \mathbb{R}^{h_2}$ is defined as

$$\mathbf{F}(\mathbf{c}) = \text{Normalize}(\text{CDIM}(\mathbf{c})). \tag{6}$$

$\text{CDIM}(\cdot)$ denotes conditional DDIM model. With Eq. (5), we can estimate the target vector in an accurate way, which is justified by Proposition 1.

Proposition 1. *For a given value of $\gamma > 0$, our method ensures that cosine similarity between target vector $\mathbf{z}$ and estimated target vector $\hat{\mathbf{z}}$ to be greater or equal than constant α with probability at least:*

$$\begin{aligned} &P(\textit{Cosine-Sim}(\mathbf{z}, \hat{\mathbf{z}}) \geq \alpha) \\ &= 1 - \int_{-1}^{\beta} \frac{\Gamma(h_2/2 + 1/2)}{\sqrt{\pi}\Gamma(h_2/2)} (1 - u^2)^{h_2/2 - 1} du \end{aligned} \tag{7}$$

where β is a scalar calculated with $\alpha, \gamma, \mathbf{c}$.

For the detailed explanation and proof, see Appendix B.

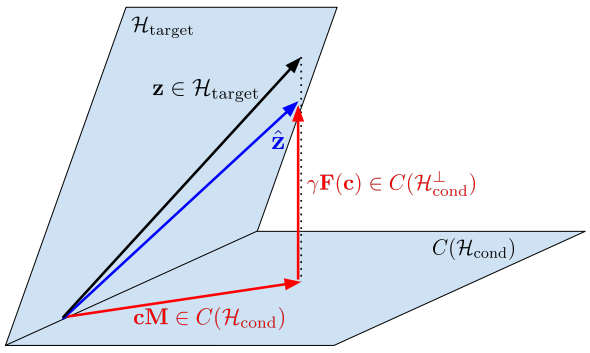


Figure 2. Sampling strategy of our model. Projection from the target hyperplane ($\mathcal{H}_{\text{target}}$) onto the column space of conditional hyperplane ($C(\mathcal{H}_{\text{cond}})$) is predicted using mapping matrix ($\mathbf{M}$), and perpendicular residuals are estimated using conditional DDIM ($\text{CDIM}(\cdot)$).

3.3. Training

In order to train our model, we used conditional DDIM loss in Eq. (2). However, DDIM loss does not guarantee guidance for the exact prediction of the latent vector. For this reason, we also use a reconstruction loss to obtain more accurate predictions:

$$\mathcal{L}_{\text{recon}} = \mathbb{E}_{\mathbf{z},\mathbf{c}}[\|\mathbf{z} - \hat{\mathbf{z}}\|_1], \tag{8}$$

Finally, our training objective is defined as a weighted sum of the DDIM (Eq. (2)) and reconstruction (Eq. (8)) losses. We optimize both $\mathbf{M}$ and q_ϕ, which are mapping matrix and parameters of conditional DDIM, respectively:

$$\begin{aligned} \min_{q_\phi, \mathbf{M}} \; &\lambda_1 \, \mathbb{E}_{\mathbf{z},\mathbf{c}}[\|\mathbf{z} - \hat{\mathbf{z}}\|_1] \\ &+ \lambda_2 \mathbb{E}_{t,\mathbf{z},\epsilon} \left[\|\epsilon - \epsilon_\theta \left(\mathbf{z}_t, t, \mathbf{c} \right) \|^2 \right], \end{aligned} \tag{9}$$

where λ_1, λ_2 are hyperparameters. Note that our model, conditional encoder-decoder, consists of mapping matrix and conditional DDIM.

Lemma 1. *Optimization of Eq. (9) guides the first term of Eq. (5) to predict a projection of target vector $\mathbf{z}$ onto the column space of the hyperplane aligned with a condition vector $\mathbf{c}$, while the second term is guided to predict a residual term which is perpendicular to the column space.*

In other words, our model learns the projection ($\mathbf{c}\mathbf{M}$) between two different hyperplanes that correspond to each modality, and the residual representation ($\mathbf{F}(\mathbf{c})$) for each condition vector, together. This process is visualized in Fig. 2, and the proof of Lemma 1 is provided in Appendix A.

4. Experiment

4.1. Datasets

We conduct main experiments on three widely-used public datasets to evaluate text-to-image generation tasks: MS-COCO [17][1], CC3M [30][2] and CelebA [18][3]. The MS-COCO dataset, released in 2014, comprises 82K training images and 40K validation images. On the other hand, the CC3M dataset contains 3.3M training examples and 16K validation examples. Unlike the MS-COCO images, which have been carefully selected, the Conceptual Captions images and their accompanying descriptions are collected from the internet, and thus, offer a broader range of styles. CelebA dataset contains 200K images of celebrities. We only use CelebA for validation, not for training.

[1]MS-COCO: https://cocodataset.org
[2]CC3M: https://ai.google.com/research/ConceptualCaptions
[3]CelebA: https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

4.2. Evaluation Protocol

For evaluation metrics, we report the Sim_{txt} and Sim_{img}, which are calculated using the validation dataset. Sim_{txt} is determined by computing the expected cosine similarity between the ground truth and predicted text features, while Sim_{img} is computed by measuring the similarity between the image and its inferred embeddings.

For the image captioning downstream task, we intend to utilize the BLEU [23] and ROUGE-L [16] evaluation metrisc. The BLEU score calculates precision of n-gram overlaps between the generated caption and the ground-truth captions. ROUGE-L measures the length of the longest common sequence among the captions. To evaluate and compare the visual quality of the generated images in the text-to-image generation task, we have chosen the Frechet Inception Distance (FID) [9] and Inception Score (IS) [29] metrics. The FID score indicates the visual similarity between the real and generated images, with lower scores being preferable. Conversely, the IS score measures the quality of the generated images, with higher scores indicating better performance. Lastly, for image classification task, we employ accuracy as the evaluation metric.

4.3. Implementation Details

Conditional encoder-decoder is implemented using the backbone of Diffusion Autoencoders [25]. For the implementation of matrix $\mathbf{M}$ and function $\mathbf{F}(\cdot)$ in Eq. (5), we use 1 FC layer with bias and modified architecture of Diffusion Autoencoders, respectively. Especially, we stacked 10 MLP layers with skip connections, and added layer normalization and dropout at the end of each layer for conditional DDIM. We trained our network with fixed learning rate of 10^{-4} and weight decay of 10^{-2}. Probability for unconditional sampling and classifier-free guidance [11] scale is set to 0.05 and 5.0, respectively. In case of batch size, we used 256 for training both $q_{\phi^{\mathbf{z}_{\text{txt}}}}$ and $q_{\phi^{\mathbf{z}_{\text{img}}}}$. Also, we empirically choose $\gamma = 1.0$, and set the value of λ_1 and λ_2 in Eq. (9) to 1.0 and 2.0, respectively.

5. Results

5.1. Modality Translation Results

Table 2, 3, 4 present the results of the image-text modality translation task on the MS-COCO [17], CC3M [30], and CelebA [18] datasets. We report cosine similarity between ground-truth and estimated target vector.

As shown in Table 2, our model outperforms all baseline models by a significant margin, and achieves better performance than VDLGAN [12] on modality translation. This result suggests that our novel and general architecture has the ability to capture relevance between two modalities better than previous methods. We also show that our method can be applied to unimodal encoders (*e.g.* BERT, DINOv2) in Table 3, proving the possibility of attacking various downstream tasks by utilizing the most powerful encoder corresponding to the particular modality. Table 4 represents cross-domain experiments, which imply that our model once trained on one dataset can be applied to another dataset without losing the representation power of latent vectors.

Dataset	MS-COCO [17]		CC3M [30]	
Method	Sim_{txt}	Sim_{img}	Sim_{txt}	Sim_{img}
LAFITE [35]	0.0965	-	0.0912	-
CLIP-GEN [34]	0.3042	-	0.2896	-
VDLGAN [12]	0.6104	0.7655	0.6237	0.7105
Ours	**0.8394**	**0.8233**	**0.7389**	**0.7443**

Table 2. Results of cosine similarity between $\mathbf{z}$ and $\hat{\mathbf{z}}$ from text, image modalities. We used CLIP ViT-B/32 [26] model for both $p_{\theta^{\mathbf{z}_{\text{txt}}}}$ and $p_{\theta^{\mathbf{z}_{\text{img}}}}$. Bold number indicates the best performance among the column and '-' indicates unavailability.

Dataset		MS-COCO [17]	
$p_{\theta^{\mathbf{z}_{\text{txt}}}}$	$p_{\theta^{\mathbf{z}_{\text{img}}}}$	Sim_{txt}	Sim_{img}
CLIP ViT-L/14 [26]	CLIP ViT-L/14	0.7765	0.8192
CLIP ViT-L/14	BERT [3]	0.9745	0.7796
DINOv2 [22]	CLIP-RNx50	0.7917	0.5207

Table 3. Results of cosine similarity between $\mathbf{z}$ and $\hat{\mathbf{z}}$ from text, image modalities using various type of latent encoders.

Dataset	MS-COCO [17] $\rightarrow$ CelebA [18]		CC3M [30] $\rightarrow$ MS-COCO	
$p_{\theta^{\mathbf{z}_{\text{txt}}}}, p_{\theta^{\mathbf{z}_{\text{img}}}}$	Sim_{txt}	Sim_{img}	Sim_{txt}	Sim_{img}
CLIP ViT-B/32 [26]	0.8237	0.5974	-	-
CLIP ViT-L/14	0.6885	0.5993	0.6817	0.7300

Table 4. Results of cross-domain experiments. We train our model on bigger dataset, and measure the cosine similarity between ground truth and predict target vector on a relatively small dataset.

5.2. Text-to-Image Generation

For the text-to-image generation task, we extract text embeddings using CLIP ViT-L/14 [26] model, then apply our model to obtain image embeddings. Finally, these embeddings are fed into a Karlo [4] decoder model to generate 256×256 images. Examples of generated images are shown in the Fig. 3 (a).

5.3. Image Retrieval

To perform image retrieval task, we use our encoder to predict an image embedding given a text embedding. Then we calculate cosine similarity of the obtained image embed-

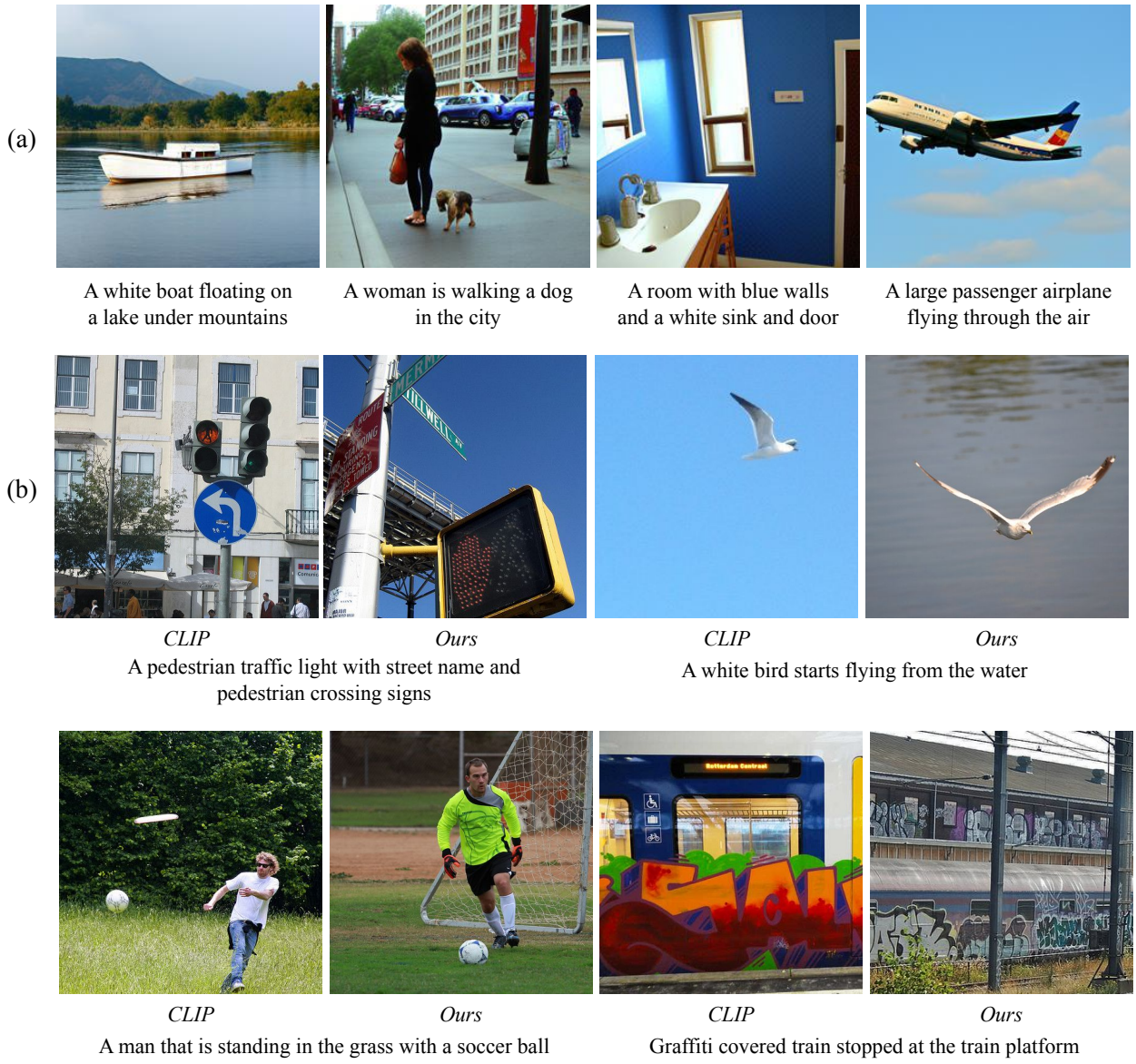


Figure 3. Results of the (a) Text-to-Image Generation, (b) Image Retrieval downstream tasks.

ding with images from the validation dataset, and retrieve an image with the highest similarity score. Fig. 3 (b) compares images obtained via our model with those retrieved using CLIP ViT-B/32 [26].

5.4. Image Captioning

After generating the text embedding with our model for a given input image, an arbitrary text decoder can translate the vector back for image captioning. In this section, we used CLIP-RNx50 [26] for the encoder and ClipCap [20] and CapDec [21] for the decoders. The results are reported in Table 5. Our model performs comparably to CapDec and slightly underperforms than ClipCap. We closely examine images from validation data in Fig. 4. While most captions are generated (a) correctly or (b) similarly, (c) some were confused about related objects such as a laptop and a mouse, and (d) some focused on different objects like man over muffin.

5.5. Image Classification

For image classification task, we utilize the CIFAR-10 dataset [14][4] as the basis for our experimentation. We em-

[4]CIFAR-10: https://www.cs.toronto.edu/~kriz/cifar.html

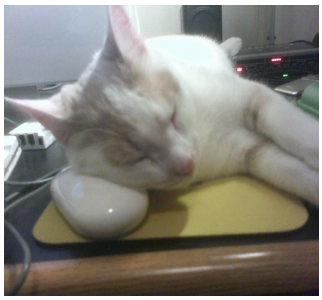
(a)

Ours + CapDec	A man holding a tennis racquet on a tennis court
Ours + ClipCap	A man holding a tennis racquet on a tennis court
References	A person holding a tennis racket in the air on a tennis court
	A man with a hat and sunglasses playing tennis
	A man holding a tennis racquet on a tennis court
	A man in sunglasses and a hat is getting ready to hit a tennis ball
	A tennis player hits the ball back to his opponent

(b)

Ours + CapDec	A little girl that is holding a toothbrush in her mouth
Ours + ClipCap	A child is brushing her teeth with a toothbrush
References	A small girl with long hair brushing her teeth
	A little girl brushing her teeth with an electric toothbrush
	a close up of a small child brushing her teeth
	A girl in pajamas brushing her teeth with an crayon toothbrush
	A little girl brushing her teeth with a tooth brush

(c)



Ours + CapDec	A cat laying on top of a laptop computer
Ours + ClipCap	A cat laying on top of a laptop
References	A cat that is laying with its head down on a mouse
	A white cat laying on the computer mouse
	A white cat is taking a nap on a mouse
	a kitty sleeping on a mouse pad and a mouse
	A cat is sleeping on a desk with its head on a computer mouse

(d)

Ours + CapDec	A man is preparing food in a kitchen
Ours + ClipCap	A person in a kitchen baking food in a oven
References	a person with a black oven mit is taking a pan out of the oven
	A person reaches into an oven to take out some muffins
	A person getting muffins out of an oven
	A man in black jacket removing tin of muffins from oven
	A muffin tray that is inside of a oven

Figure 4. Results of the Image Captioning downstream task.

ploy the CLIP VIT-L/14 [26] and CLIP ViT-B/32 models to extract class embeddings in the form of "a picture of a {class name}". Furthermore, we extract image embeddings using CLIP and subsequently apply our proposed model to obtain text embeddings. Finally, by comparing the text embeddings from the images with the class embeddings, we calculate the cosine similarity and derive accuracy metrics for the image classification task. Classification results are reported in Table 6.

Method	B@1	B@4	R-L
ClipCap [20]	**74.7**	**33.5**	-
CapDec [21]	69.2	26.4	**51.8**
Ours + ClipCap	65.9	23.6	47.7
Ours + CapDec	67.7	25.5	48.7

Table 5. Results for image captioning on MS-COCO dataset.

Method	Accuracy (%)
CLIP ViT-L/14 [26]	**85.05**
CLIP ViT-B/32	**69.69**
Ours + CLIP ViT-L/14	77.11
Ours + CLIP ViT-B/32	60.74

Table 6. Results for image classification on CIFAR-10 dataset.

Dataset	MS-COCO [17]		MS-COCO $\rightarrow$ CelebA [18]	
Method	Sim_{txt}	Sim_{img}	Sim_{txt}	Sim_{img}
w.o / Projection	0.6280	0.6155	-	-
w.o / DDIM	0.8386	0.8199	0.8145	0.5922
Ours	**0.8394**	**0.8233**	**0.8237**	**0.5974**

Table 7. Ablation study on our conditional encoder-decoder model. Note that "w/o Projection" means only using the second term in Eq. (5), while "w/o DDIM" corresponds to using only the first term.

5.6. Ablation Study

We report our ablation study results in Table 7. Ablation study shows that using both projection matrix and conditional DDIM in Eq. (5) gives the best performance for modality translation for both in-domain and cross-domain experiments.

6. Conclusion

We presented a diffusion-based encoder-decoder architecture for translation between different modalities. Our model achieves highest similarity between the image and text modalities on several datasets, with arbitrary encoders and decoders attached to our model. We also conducted experiments on various downstream tasks and demonstrated effectiveness and versatility of our model. Our model can be further extended to sound modality though there are no publicly available encoder and decoder models for audio dataset currently to the best of our knowledge. Additionally, because our model is based on diffusion, the overall training takes relatively long compared to other models. We leave the exploration of injecting lightweight models or adopting efficient diffusion methods for future studies.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. 2

[2] Eungchun Cho. Inner product of random vectors. *International Journal of Pure and Applied Mathematics*, 56, 01 2009. i

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1, 2, 5

[4] Jisu Choi Jongmin Kim Minwoo Byeon Woonhyuk Baek Donghoon Lee, Jiseob Kim and Saehoon Kim. Karlo-v1.0.alpha on coyo-100m and cc15m. https://github.com/kakaobrain/karlo, 2022. 5

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 3

[6] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12868–12878, 2020. 2

[7] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *ArXiv*, abs/2104.08821, 2021. 2

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[9] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 5

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2

[11] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2, 3, 5

[12] Minsoo Kang, Doyup Lee, Jiseob Kim, Saehoon Kim, and Bohyung Han. Variational distribution learning for unsupervised text-to-image generation. *ArXiv*, abs/2303.16105, 2023. 5

[13] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2018. 2

[14] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 6

[15] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, 2022. 2, 3

[16] Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Annual Meeting of the Association for Computational Linguistics*, 2004. 5

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 4, 5, 8

[18] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 4, 5, 8

[19] Zheyuan Liu, Cristian Rodriguez-Opazo, Damien Teney, and Stephen Gould. Image retrieval on real-life images with pre-trained vision-and-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2125–2134, 2021. 2

[20] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021. 2, 3, 6, 8

[21] David Nukrai, Ron Mokady, and Amir Globerson. Text-only training for image captioning using noise-injected clip. In *Conference on Empirical Methods in Natural Language Processing*, 2022. 2, 3, 6, 8

[22] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1, 2, 5

[23] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics*, 2002. 5

[24] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. *ArXiv*, abs/2302.03027, 2023. 2

[25] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10619–10629, 2022. 1, 3, 5

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 1, 2, 3, 5, 6, 7, 8

[27] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125, 2022. 2

[28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 2

[29] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 5

[30] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018. 4, 5

[31] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[32] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2

[33] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 3

[34] Zihao Wang, Wei Liu, Qian He, Xinglong Wu, and Zili Yi. Clip-gen: Language-free training of a text-to-image generator with clip. *arXiv preprint arXiv:2203.00386*, 2022. 2, 5

[35] Yufan Zhou, Ruiyi Zhang, Changyou Chen, Chunyuan Li, Chris Tensmeyer, Tong Yu, Jiuxiang Gu, Jinhui Xu, and Tong Sun. Towards language-free training for text-to-image generation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17886–17896, 2021. 1, 2, 5

Appendix

A. Proof of Lemma 1.

Lemma 1. *Optimization of Eq. (9) guides the first term of Eq. (5) to predict a projection of target vector* $\mathbf{z}$ *onto the column space of the hyperplane aligned with a condition vector* $\mathbf{c}$*, while the second term is guided to predict a residual term which is perpendicular to the column space.*

Proof. Let $\mathbf{z} \in \mathcal{H}_{\text{target}}$ and $\mathbf{c} \in \mathcal{H}_{\text{cond}}$, where $\mathcal{H}_{\text{target}}$ and $\mathcal{H}_{\text{cond}}$ denote a hyperplane aligned with the target and condition vectors, respectively. Recall that our method's reconstruction loss is defined as

$$\mathbb{E}_{\mathbf{z},\mathbf{c}}[\|\mathbf{z} - \hat{\mathbf{z}}\|_1], \tag{10}$$

and for the proper value of hyperparameter γ such that $\gamma \geq 1/2\|\mathbf{z} - \mathbf{cM}\|_1$, inequality

$$\mathbb{E}_{\mathbf{z},\mathbf{c}}[\|\mathbf{z} - \hat{\mathbf{z}}\|_1] \geq \mathbb{E}_{\mathbf{z},\mathbf{c}}[\|\mathbf{z} - \mathbf{cM}\|_1] \tag{11}$$

holds. So minimizing the reconstruction loss in Eq. (10) induces the optimization of the right term of Eq. (11). Recall that in the least squares,

$$\frac{\partial}{\partial \mathbf{M}}\|\mathbf{Z} - \mathbf{CM}\|^2 = 2\mathbf{C}^\top(\mathbf{CM} - \mathbf{Z}) \tag{12}$$

where $\mathbf{C}, \mathbf{Z}$ are the matrices corresponding to $\mathcal{H}_{\text{target}}, \mathcal{H}_{\text{cond}}$, and the solution of Eq. (12) is

$$\hat{\mathbf{M}} = (\mathbf{C}^\top\mathbf{C})^{-1}\mathbf{C}^\top\mathbf{Z}. \tag{13}$$

Then we define the *Projection matrix* $\mathbf{P}$ as

$$\mathbf{P} = \mathbf{C}(\mathbf{C}^\top\mathbf{C})^{-1}\mathbf{C}^\top, \tag{14}$$

and $\mathbf{c}\hat{\mathbf{M}} = \mathbf{Pz}$ is a vector on the column space of $\mathcal{H}_{\text{cond}}$, denoted as $C(\mathcal{H}_{\text{cond}})$.

From the optimization of Eq. (12), when loss terms are fully optimized, we can assume that $\mathbf{M} = \hat{\mathbf{M}}$ and so

$$\mathbf{cM} \in C(\mathcal{H}_{\text{cond}}). \tag{15}$$

For each data point, there is a residual vector defined as $\hat{\mathbf{z}} - \mathbf{cM}$, and we model this using conditional diffusion model, denoted as $\gamma\mathbf{F}(\mathbf{c})$. With the assumption used in Eq. (15), we can derive Eq. (16),

$$\gamma\mathbf{F}(\mathbf{c}) \in C(\mathcal{H}_{\text{cond}})^\perp. \tag{16}$$

□

B. Proof of Proposition 1.

Proposition 1. *For a given value of* $\gamma > 0$*, our method ensures the cosine similarity between target vector* $\mathbf{z}$ *and estimated target vector* $\hat{\mathbf{z}}$ *to be greater or equal than constant* α *with probability at least:*

$$P(\textit{Cosine-Sim}(\mathbf{z}, \hat{\mathbf{z}}) \geq \alpha) = 1 - \int_{-1}^{\beta} \frac{\Gamma(h_2/2 + 1/2)}{\sqrt{\pi}\Gamma(h_2/2)}(1 - u^2)^{h_2/2-1}du \tag{17}$$

where β *is a scalar calculated with* $\alpha, \gamma, \mathbf{c}$*.*

Proof. Using Eq. (15) and (16), cosine similarity between $\mathbf{z}$ and $\hat{\mathbf{z}}$ is calculated as:

$$\begin{aligned}
&\text{Cosine-Sim}(\mathbf{z}, \hat{\mathbf{z}}) \\
&= \mathbf{z} \cdot \text{Normalize}(\mathbf{cM} + \gamma\mathbf{F}(\mathbf{c})) \\
&= \mathbf{z}^\top \frac{\mathbf{cM} + \gamma\mathbf{F}(\mathbf{c})}{\|\mathbf{cM} + \gamma\mathbf{F}(\mathbf{c})\|_2} \\
&= \frac{\mathbf{z}^\top\mathbf{cM} + \gamma\mathbf{z}^\top\mathbf{F}(\mathbf{c})}{\sqrt{(\mathbf{cM})^\top\mathbf{cM} + 2\gamma\mathbf{zMF}(\mathbf{c}) + \gamma^2\|\mathbf{F}(\mathbf{c})\|_2^2}} \\
&= \frac{\mathbf{z}^\top\mathbf{cM} + \gamma\mathbf{z}^\top\mathbf{F}(\mathbf{c})}{\sqrt{(\mathbf{cM})^\top\mathbf{cM} + \gamma^2\|\mathbf{F}(\mathbf{c})\|_2^2}},
\end{aligned} \tag{18}$$

Using the property of projection matrix, Eq. (19) and (20) holds,

$$\mathbf{z}^\top\mathbf{cM} = \mathbf{z}^\top\mathbf{Pz} = \mathbf{z}^\top\mathbf{P}^\top\mathbf{Pz} = (\mathbf{Pz})^\top\mathbf{Pz}, \tag{19}$$

$$\mathbf{cM} = \mathbf{Pz}, \tag{20}$$

and the cosine similarity is simplified as

$$\begin{aligned}
\text{Cosine-Sim}(\mathbf{z}, \hat{\mathbf{z}}) &= \frac{(\mathbf{Pz})^\top\mathbf{Pz} + \gamma\mathbf{z}^\top\mathbf{F}(\mathbf{c})}{\sqrt{(\mathbf{Pz})^\top\mathbf{Pz} + \gamma^2\|\mathbf{F}(\mathbf{c})\|_2^2}} \\
&= \frac{\sum_i {a_i}^2 + \gamma\mathbf{z}^\top\mathbf{F}(\mathbf{c})}{\sqrt{\sum_i {a_i}^2 + \gamma^2}},
\end{aligned} \tag{21}$$

Note that $\{a_i\}$ in Eq. (21) are defined as the coefficients of linear combination of $\{\mathbf{e}_i\}$ to represent $\mathbf{Pz}$, where $\{\mathbf{e}_i\}$ are defined as orthonormal basis of $C(\mathcal{H}_{\text{cond}})$.

Using the cumulative distribution function of the inner product of random vectors $\mathbf{x}, \mathbf{y}$ on the sphere derived in [2], we can derive:

$$P(\mathbf{x}^\top\mathbf{y} \leq z) = \int_{-1}^{z} \frac{\Gamma(d/2 + 1/2)}{\sqrt{\pi}\Gamma(d/2)}(1 - u^2)^{d/2-1}du, \tag{22}$$

where d is the dimension of a random vector, and $\Gamma(x)$ is gamma function defined as $\Gamma(x) = \int_0^\infty u^{x-1}e^{-u}du$.
Combining Eq. (21) and (22), we can derive that for a given value of $\gamma > 0$, our method ensures the cosine similarity

between original latent vector $\mathbf{z}$ and predicted latent vector $\hat{\mathbf{z}}$ to be greater or equal than constant α with probability at least:

$$P(\text{Cosine-Sim}(\mathbf{z}, \hat{\mathbf{z}}) \geq \alpha) = 1 - \int_{-1}^{\beta} \frac{\Gamma(h_2/2 + 1/2)}{\sqrt{\pi}\Gamma(h_2/2)} (1 - u^2)^{h_2/2-1} du \quad (23)$$

where $\beta = (\alpha\sqrt{\sum_i {a_i}^2 + \gamma^2} - \sum_i {a_i}^2)/\gamma$, ahd h_2 is a dimension of target vector.

□