An Image is Worth One Sentence: Fast Textual Inversion with Supreme Intialization

Haojun Qiu, Zixin Guo, Zixin Wei
University of Toronto
{harry.qiu, zixin.guo, jenci.wei}@mail.utoronto.ca

Abstract

Text-driven image synthesis has emerged as a popular research area. Existing approaches to invert image to text face challenges like requiring multiple images, slow convergence, or overfitting thus suffering from editing capability. In this paper, we propose a novel initialization method for inverting text using off-the-shelf classification or captioning models. This approach enables multi-token embedding learning from a single input image while eliminating the need for fine-tuning and ensuring faster convergence. We demonstrate a significant improvement in convergence speed compared to vanilla TI.

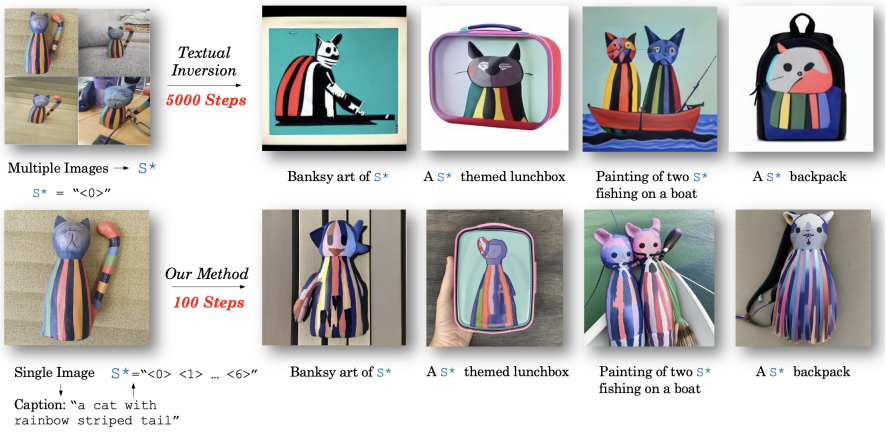


Figure 1: Inverting the input image(s) (left) to generate personalized outputs (right). We learned that text embeddings from single image can do plausible personalization, converging much faster than TI.

1 Introduction

In recent years, text-driven image synthesis has garnered attention for its human-level easy conditioning. However, generating a specific concept or object, like your own cat, is not controllable by providing only our prompt description, thus inverting image to text becomes a popular research area. The work Textual Inversion (TI) ([1]) utilizes continuous optimization to convert image concepts into text embeddings, enabling personalization when combined with a prompt template. However, TI often demands multiple images for optimal text embeddings, which can be costly and impractical. Additionally, convergence speed and effectiveness depend largely on quality of initialization tokens. While there are some methods resolve the two issues above, i.e., using single image achieving

Preprint. Under review.

relatively fast convergence, but they fine-tune the pre-trained diffusion model, overfitting to input image and harming editing capabilities.

To overcome the challenges mentioned, we proposed to supremely initialize the optimizing embeddings using classification or captioning, which are considered as "problem solved" high-level vision applications. By doing so, we need to optimize for multiple embeddings (depends on the class label/caption length), unlike the restriction of single embedding imposed in vanilla TI. Moreover, we align with the vanilla TI to not perform any fine-tuning, which is memory efficient and the optimization process is less complex. We experimentally found that with our initialization scheme, even given only one image, we gain a convergence speed orders of maginitude faster compared with the vanilla TI with embedding of token '$*$' as initialization, as shown in Figure 3.

In summary, our contributions are of several folds: (1) we performed *multi-tokens embeddings* learning for representing a concept from a *single* input image; (2) we proposed *a mindset of initialization scheme* for inverting text, leveraging state-of-the-art classification and captioning models; (3) we *evaluated* that by using our initialization scheme, we *dramatically speeds up the convergence* in comparison to vanila textual inversion.

2 Related Work

Text-guided image synthesis. Text-guided image synthesis has been extensively explored in the context of generative adversarial networks (GANs) ([2, 3, 4, 5]), but challenges such as mode collapse and convergence instability persist, limiting scalability. Recently, diffusion models ([6, 7, 8]) have demonstrated high-quality image generation by learning to denoise images. This has spurred investigations into conditional synthesis, particularly when conditioned on text. DALLE-2 ([6]) utilizes CLIP's guidance ([9]) for image generation. Latent Diffusion Models (LDMs) ([10]) enhance the efficiency by moving computation to a latent space. Our work builds on these pre-trained text-conditional image synthesis models, particularly LDMs, and we focus on determining an effective text embedding for a single input image that enables personalization.

Inverting Text and Personalization. Various image editing and personalization approaches exist, each facing some challenges. Methods like Textual Inversion (TI) ([1]), DreamBooth ([11]) require multiple input images of the same concept, rendering them costly and impractical. On the other hand, lots of methods fine-tunes the model weight ([11, 12, 13, 14]), which can be memory-intensive due to saving a copy of model for each (set of) input image. Moreover, some approaches ([12, 13, 15]), are limited to strict editing, constraining output diversity and personalization. Some work tries to map the entire image into the text/embedding space ([16]), however, we are only interested in the object of focus. From the above, TI is the most relevant work as it only optimizes token embeddings. Additionally, we receives only a single image as input and has a supreme multi-token initialization. We demonstrate that an effective initialization scheme for multi-embeddings learning significantly accelerates convergence.

Image Classification and Captioning. We investigate two methods for initializing textual inversion tokens using state-of-the-art classification and captioning models. Vision Transformers ([17]) are employed for classification, while Caption Anything ([18]), which integrates Segment Anything ([19]) and ChatGPT ([20]), enable context-aware captioning. Users can identify object locations for custom captions. Both approaches are vital for initializing textual inversion tokens before optimization.

3 Method

We first briefly outline the textual inversion method ([1]) that uses latent diffusion model ([10]). Then, we explain our proposed mutli-token representations and initialization scheme.

Latent Diffusion Model. We implement our method using the latent diffusion model (LDM), which has two main components. One is a variational auto-encoder ([21]) pre-trained on a large collection of images. Given an image $\mathbf{x}$, we obtain its latent code / feature map $\mathbf{z} = \mathcal{E}(\mathbf{x})$ for encoder $\mathcal{E}$, and can transform latent code back to image space using $\tilde{\mathbf{x}} = \mathcal{D}(\mathbf{z})$ for decoder $\mathcal{D}$. To train it, the common VAE training objective is applied, and the diffusion training process is done in the

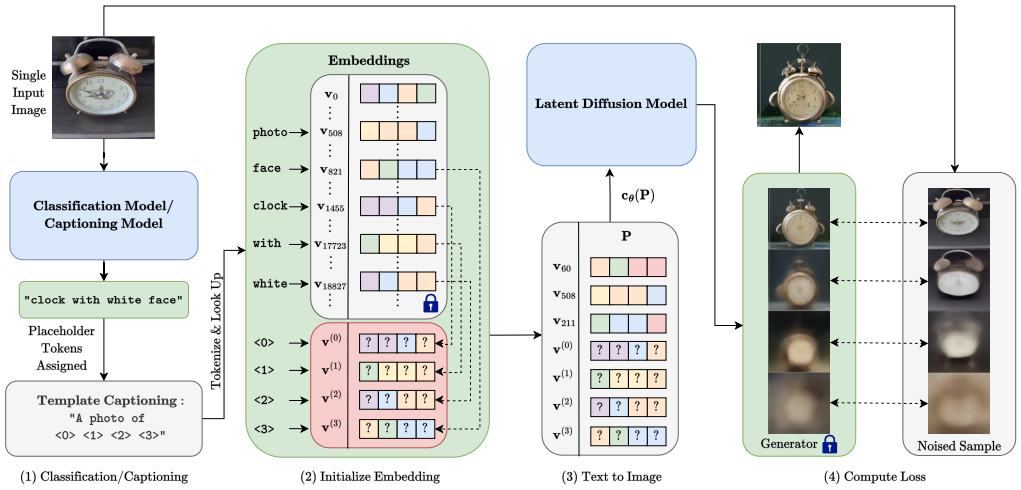


Figure 2: **Algorithm Diagram Box**. (1) The input image is first classified/captioned, and the template containing placeholder tokens is generated to match captioning length. (2) The template is tokenized and converted to a matrix $\mathbf{P}$, each row corresponds to a word embedding. The dotted lines show that each placeholder token is initialized to their corresponding output from the classification/captioning model. We only optimize parameters in the red region. (3) The matrix $\mathbf{P}$ is converted into a vector by text-encoder $\mathbf{c}_{\boldsymbol{\theta}}(.)$ then fed into the LDM generator. (4) Compute the LDM loss. The generated image and noised samples shown are for illustration, loss is computed in *latent space* as in (1)

latent space of the auto-encoder instead of the image pixel space. Similar to DDPM ([22]), the noise prediction module in the latent space receives a uniformly sampled time step t and a noisy latent code $\mathbf{z}_t$ computed by adding scheduled noise to $\mathbf{z} = \mathcal{E}(\mathbf{x})$ for some image $\mathbf{x}$. In addition, it can receive a conditional vector $\mathbf{c}_{\boldsymbol{\theta}}(\mathbf{y})$ by transforming some raw modality input (e.g., text) $\mathbf{y}$ using module $\mathbf{c}_{\boldsymbol{\theta}}$ (e.g., a text encoder). Overall, the training objective for LDM is

$$\mathcal{L}_{\text{LDM}} = \mathbb{E}_{\mathbf{z}\sim\mathcal{E}(\mathbf{x}),\mathbf{y},\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}),t}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}_t, t, \mathbf{c}_{\boldsymbol{\theta}}(\mathbf{y}))\|_2^2\right]. \tag{1}$$

Textual Inversion using LDM. We create a placeholder token `<s>`, and corresponds it to a newly initialized embedding vector $\mathbf{v}$, where this initialization can be random or a copy of an existing token's embedding. We will place the placeholder token into several templates to complete the prompts, e,g, `"A photo of <s>"`, `"A rendition of <s>"`, etc. Let matrix $\mathbf{P}$ denotes such prompt's embedding, i.e., each row is a token embedding so $\mathbf{v}$ is one of such row. The goal is to minimize the reconstruction loss when feeding a noise vector and the prompt embedding into the denoising module, by optimizing the embedding vector $\mathbf{v}$, i.e., we want to find

$$\mathbf{v}^* = \underset{\mathbf{v}}{\text{argmin}}\ \mathbb{E}_{\mathbf{z}\sim\mathcal{E}(\mathbf{x}),\mathbf{P},\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}),t}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}_t, t, \mathbf{c}_{\boldsymbol{\theta}}(\mathbf{P}))\|_2^2\right].$$

This is pure optimization for embedding vector $\mathbf{v}$, all the pre-trained model's weights are frozen.

Multi-Token Representation. Instead of representing a concept using only one *single* "new word", i.e., a placeholder token `<s>` as in the vanilla TI, we allow constructing *multiple* placeholder tokens `<0>, <1>, ..., <n>`, and denote their corresponding optimizing embeddings as $\mathbf{v}^{(0)}, \mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n)}$). The specific number of placeholder tokens n that we need to add/initialize is the number of tokens in the classification/captioning model outputs, which we discuss in next section. During training, the embeddings of these placeholder tokens are optimized jointly to better represent the desired concept in the input image. After training, similar to vanilla TI, we generate a prompt by placing the multiple placeholder tokens (whose embeddings has been optimized already) separated by spaces in order, into the template. The intuition for having multiple tokens is to increase the expressiveness for the model to capture the desired concept, thus represents it much more accurately.

Supreme Initialization. We propose two methods for initializing the placeholder token of an input image. These methods utilize off-the-shelf, state-of-the-art pre-trained models. The first method

involves feeding the input image into our image classification model and using its output to initialize the placeholder tokens. The second method utilizes captioning output as initialization. Specifically, when provided with a single input image into our image captioning model, we can click on the object of interest within the image. The image captioning model generates a caption based on the clicked region, which is then employed as the initialization embeddings for multiple placeholder tokens. The goal of these initialization methods is to provide the model with a strong starting point, enabling it to converge faster to the true word embeddings that represents the concept of interest in the text space.

4 Discussion

In the following section, we evaluate and compare the convergence of textual inversion training under different setups. For more details about dataset, experiment set up, and results, please refer to the appendix. In addition, an ablation study of the two proposed methods is included in Section A.4.

4.1 Results

We use three metrics to evaluate our proposed model's performance, each focusing on different aspects. We first log the LDM losses, which is the MSE between the noise prediction and ground truth noise in the latent space as in Equation (1), during the training at every step. This is the most direct way to evaluate how the embeddings that represent the concept is learned from the input image. The other metrics are LPIPS ([23]) and SIFID ([24]), both assess the quality of the generated images in terms of human perception, further explained in Section A.3.

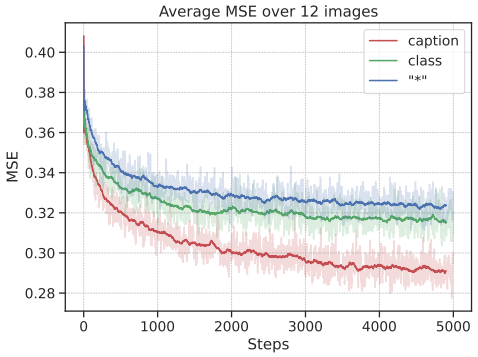


Figure 3: The plot illustrates the training average mean squared error, i.e., the LDM loss over 12 images. The red curve shows using caption as initialization, the green curve shows using object class as initialization, and blue curve shows '$*$' as initialization.

For each of the three initialization settings: '$*$', object class, and caption, we evaluate the MSE loss averaged over 12 input images of different categories. As seen from Figure 3, the MSE for caption initialization (which usually contains more tokens than classification output and is a more detailed description) converges much faster than the other methods, matching our intuition. The loss for object class initialization converges faster than that of '$*$' initialization. These results demonstrate the effectiveness of our proposed initialization scheme in accelerating convergence and enhancing the overall performance of textual inversion. The evaluations using the metrics LPIPS and SIFID, shown in Section A.3, also agree with our findings.

4.2 Limitation and Next Steps

Firstly, we only evaluated our initialization scheme when given single input image. However, we can run all the experiments in the setting of multiple images to see even more directly how much improvements our method gains in comparison to vanilla textual inversion. Secondly, we ran the experiments on a set of 12 images, but more images help to generalize the conclusions made.

5 Conclusion

In this work, we present multi-tokens embeddings learning, which effectively represents the concept of one image. Using which, we also proposed a novel initialization scheme for inverting text that leverages state-of-the-art classification and captioning models, bridging the gap between image and textual representations. Our evaluation results demonstrated that our initialization scheme dramatically speeds up the convergence when compared to vanilla textual inversion, making it a promising approach for future research and applications.

References

[1] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion, 2022.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[3] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5802–5810, 2019.

[4] Ming Tao, Hao Tang, Songsong Wu, Nicu Sebe, Xiao-Yuan Jing, Fei Wu, and Bingkun Bao. Df-gan: Deep fusion generative adversarial networks for text-to-image synthesis. *arXiv preprint arXiv:2008.05865*, 2020.

[5] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324, 2018.

[6] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022.

[7] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[8] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[9] 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, 2021.

[10] 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.

[11] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. 2022.

[12] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. *arXiv preprint arXiv:2210.09276*, 2022.

[13] Dani Valevski, Matan Kalman, Yossi Matias, and Yaniv Leviathan. Unitune: Text-driven image editing by fine tuning an image generation model on a single image. *arXiv preprint arXiv:2210.09477*, 2022.

[14] Rinon Gal, Moab Arar, Yuval Atzmon, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. Encoder-based domain tuning for fast personalization of text-to-image models, 2023.

[15] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. *arXiv preprint arXiv:2211.09794*, 2022.

[16] Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery, 2023.

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[18] Teng Wang, Jinrui Zhang, Junjie Fei, Zhe Li, Yunlong Tang, Mingqi Gao, and Zheng Hao. Caption-anything. *https://github.com/ttengwang/Caption-Anything*, 2023.

[19] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[20] OpenAI. Chatgpt: a large language model. `https://openai.com/`, 2023.

[21] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[23] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[24] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4570–4580, 2019.

[25] Unsplash. https://unsplash.com/.

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

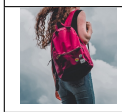
[27] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[28] 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, 2018.

A Supplementary Materials

A.1 Dataset and Classification/Captioning Model Output

The images in the dataset were either taken from the authors of the Dreambooth/Textual Inversion or obtained from Unsplash ([25]). Below we demonstrate their initialization from classification/captioning model output.

Images	Classification Output	Captioning Output
	coffee mug	a skull and bones mug
	cup	bon appetit
	sunglasses	a pair of sunglasses with a gold frame and brown lenses
	backpack	hershel supply co backpack
	sneaker	a white and pink sneaker with holographic
	teddy bear	a red plush toy with eyes and legs
	bird	a sculpture of bird
	go-kart	a toy car with a boy in it
	maraca	a cat with a rainbow striped tail
	analog clock	a clock with a crown on the face
	teapot	a red tea kettle with gold leaf
	ocarina	the elephant is carved in a natural stone

A.2 Some Personalization Result

The Fig. A.2 illustrates the generated images from prompts with learned embeddings. The object embeddings are optimized for only 100 steps and the results show promising personalization capabilities of our proposed method.

A.3 Additional Evaluation Metrics for Concept Learned

Additional to the training MSE, we further employed LPIPS and SIFID as evaluation metrics to obtain a comprehensive evaluation of how the concept are learned, i.e., the ability to reconstruct the image with desired concept using the learned embedding. Those metrics consider the perceptual similarity, and internal patch statistics, respectively.

Learned Perceptual Image Patch Similarity (LPIPS). This metric ([23]) focuses on evaluating the perceptual similarity between two images. It leverages a pre-trained VGG ([26]) to compute the similarity between feature maps at various intermediate layers of the model, i.e., which strikes a balance between high-level semantics and low-level textures. We evaluated this metric between generated image and the single input image, averaging over several generated images, where the generated images come from the reconstruction-oriented prompts below:

- `"A photo of <0> ... <n>"`
- `"A photo of a <0> ... <n>"`
- `"A good photo of a <0> ... <n>"`

The results of evaluation using LPIPS is shown in Figure 4 (lower is better).

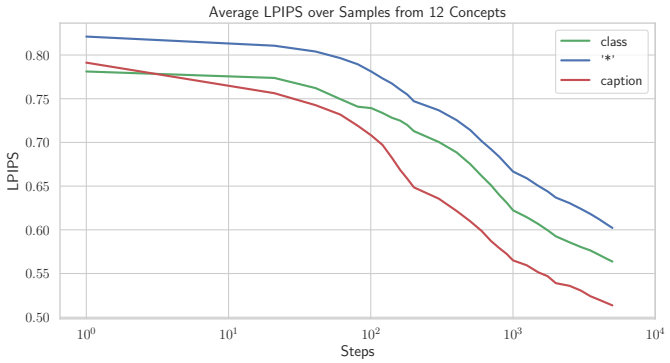


Figure 4: The figure illustrates the LPIPS for (1) "$*$" as init token (2) object class as init token (3) image caption (our proposed) as init token over 12 image concepts

Single Image Fréchet Inception Distance (SIFID). In addition to assessing the feature maps over all different layers of the feature extractor, we use the evaluation metric SIFID proposed by SinGAN ([24]), estimating the matching of the patch statistics between two images. Using the Inception Network V3 ([27]), we evaluate the FID score ([28]) between the feature layer just before the second pooling layer. This metric is particularly useful for evaluation in the single image generation setting. The generated images is acquired the same way as in evaluating LPIPS. The results of evaluation using SIFID is shown in Figure 5 (lower is better).

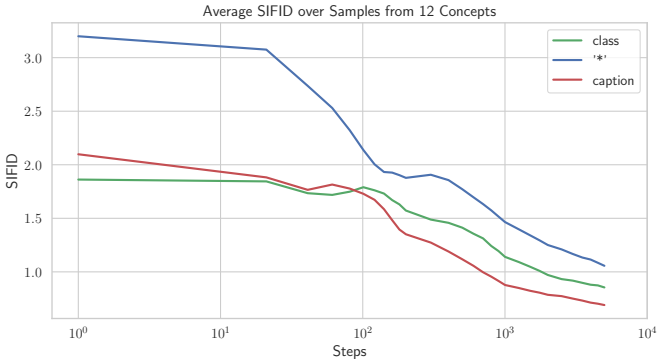


Figure 5: The figure illustrates the Single-Image FID for (1) "$*$" as init token (2) object class as init token (3) image caption as init token over 12 image concepts

A.4 Ablation Studies

We proposed both multi-tokens representation and the weight initialization scheme. We evaluated that both of the methods help with convergence in general. Specifically, we compare the performance of three set-ups (1) using the semantically meaningless '$*$' as single embedding initialization, (2) initializing the same number of tokens as corresponding image's captioning output, all with '$*$', (3) using the captioning as initialization for multiple tokens. Observing from the Figures 6, 7, 8, we make two observation and conclusions below:

- More tokens are more expressive in describing the concept in the image. This can be seen from the fact that initialization with caption and multiple '$*$' giving a lower loss, as well as them having a high convergence speed. Also, they have a better trend to converge.
- Caption initialization (which has multiple tokens) yields a better starting point than initialization with multiple '$*$', albeit the two curves meet at later iterations. Since personalization is done well before two loss curves meet (to prevent from overfitting), caption initialization can be more quickly trained in order to perform personalization.

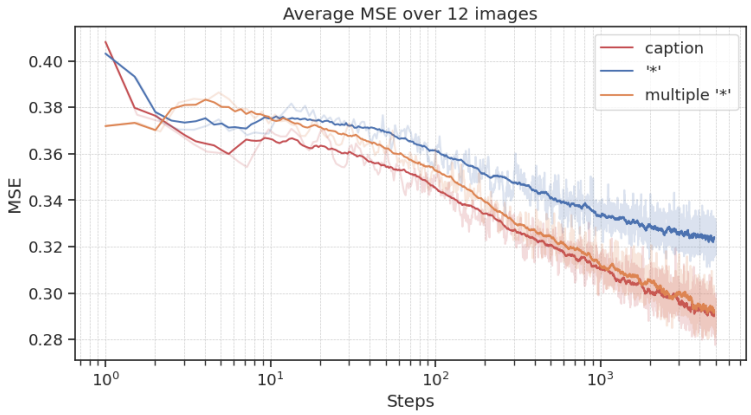


Figure 6: The figure illustrates the MSE for (1) "*" as init token, (2) multiple "*" as init token, and (3) caption (our proposed) over 12 image concepts

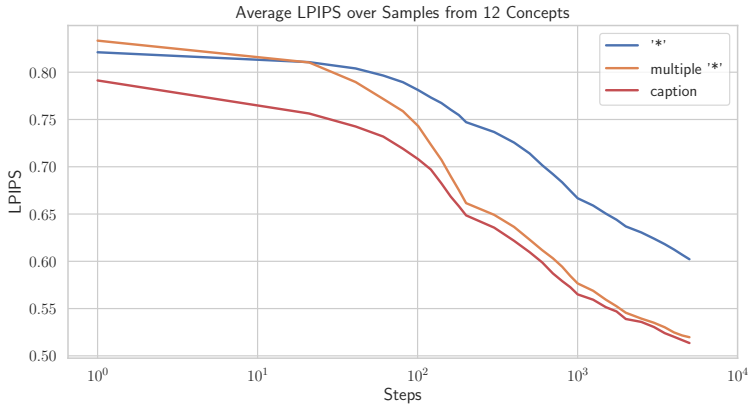


Figure 7: The figure illustrates the LPIPS for (1) "$*$" as init token (2) multiple "$*$" as init token (3) image caption (our proposed) as init token over 12 image concepts

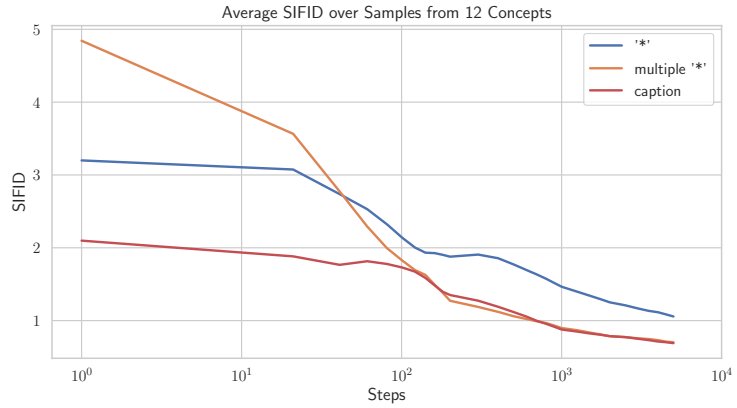


Figure 8: The figure illustrates the Single-Image FID for (1) "$*$" as init token (2) multiple "$*$" as init token (3) image caption as init token over 12 image concepts