VETIM: Expanding the Vocabulary of Text-to-Image Models only with Text

Martin Nicolas Everaert[1]
martin.everaert@epfl.ch
Marco Bocchio[2]
marco.bocchio@largo.ai
Sami Arpa[2]
sami.arpa@largo.ai
Sabine Süsstrunk[1]
sabine.susstrunk@epfl.ch
Radhakrishna Achanta[1]
radhakrishna.achanta@epfl.ch

[1] School of Computer and Communication Sciences, École polytechnique fédérale de Lausanne (EPFL), Station 14, 1015, Lausanne, Switzerland

[2] Largo.ai
EPFL Innovation Park, Building I
1015, Lausanne, Switzerland

Abstract

Text-to-image models, such as Stable Diffusion, can generate high-quality images from simple textual prompts. With methods such as Textual Inversion, it is possible to expand the vocabulary of these models with additional concepts, by learning the vocabulary embedding of new tokens. These methods have two limitations: slowness of optimisation and dependence on sample images. Slowness mainly stems from the use of the original text-to-image training loss, without considering potential auxiliary supervision terms. Relying on sample images enables learning new visual features but restricts the vocabulary expansion to concepts with pre-existing images. In response, we introduce a novel approach, named VETIM, which takes only a textual description of the concept as input. It expands the vocabulary through supervision only at the text encoder output, without accessing the image-generation part, making it faster at optimisation time. It also does not copy visual features from sample images. Our method can be used directly for applications that require a concept as a single token but do not require learning new visual features. Our approach shows that a mere textual description suffices to obtain a single token referring to a specific concept. To show the effectiveness of our method, we evaluate its performance subjectively and through objective measures. The results show that our approach is effective in expanding the vocabulary of text-to-image models without requiring images.

1 Introduction

Recent text-to-image generation models have achieved remarkable results in creating images from natural language prompts [2, 3, 11, 15, 16, 17, 18]. Once trained though, these models have a fixed vocabulary. Approaches like Textual Inversion [4, 9] can expand this vocabulary by learning the embedding of a new token representing a concept.

However, such approaches use the original text-to-image training loss, for instance, the diffusion loss [4], to optimise the embedding of the new token. This requires forward and backward passes through the image-generation part of the model, namely, the denoising U-Net for Stable Diffusion [17], making vocabulary expansion slow and computationally expensive. Also, quite naturally, such approaches require sample images of the concept as input. Requiring images for a text-to-image model is paradoxical. In many situations, one only has a textual description of the concept and, not pre-existing images.

In this paper, we consider an alternative loss term consisting of supervising the optimisation at the output of the text encoder. We show that this supervision alone is sufficient for text-based vocabulary expansion of text-to-image models, extending the vocabulary of the text-to-image models from textual descriptions only, without the need for sample images, and without accessing the image-generation part of the model at all. Unlike image-based vocabulary expansion [4, 9], our text-based vocabulary expansion learns an embedding that represents a concept described using text input only, rather than learning new visual attributes contained in sample images. We name our technique VETIM, short for *Vocabulary Expansion in Text-to-Image Model*, and describe it in Section 3. As demonstrated in Figure 1 and Section 4, our embeddings are often on par with using the full-text description to generate images. This provides the advantages of using a single token, including, for instance, overwriting the embedding of a biased token with an unbiased one to prevent misuse of a model, and other applications described in Section 5. Thanks to supervision solely from the text encoder, VETIM is efficient and fast, taking around two minutes, compared to one hour [7] for Textual Inversion [4]. Our findings also underscore the adaptability of the image-generation module, as it successfully generalises to embeddings learned exclusively with supervision at

Figure 1: **VETIM** learns the embedding of a single token, referred to here as S_*. This customised token can be used in place of a long text description, which in this case is: "a small, brilliant red stone that can produce the Elixir of Life and turn base metals into gold". Images are generated with Stable Diffusion [17], on the top using the lengthy description, and on the bottom using our token S_*. We use the same seeds for the top and bottom rows. Our token S_* appears as effective as the lengthy description. Notably, the vocabulary embedding of S_* is rapidly optimised solely from text input, without accessing the image-generation module and without copying visual attributes from any sample image.

the text encoder output. Our main contributions are thus as follows:

- We propose to learn new vocabulary embeddings using only text descriptions as input, and with supervision only at the text encoder's output. This supervision is faster than the original text-to-image training loss, does not mimic any sample images, and allows for expanding the model's vocabulary also when images of the concept are unavailable.
- Our method enables the use of text descriptions as single tokens, enabling applications otherwise limited to the word level, like debiasing and word-swapping.
- A key finding is that Stable Diffusion [17] is able to generalise to embeddings learned without supervision from the model itself. Indeed, VETIM only uses supervision from the text encoder to learn an embedding. The training signal from supervising at the text encoder level provides much faster convergence than the training signal from the original text-to-image model training loss.

2 Related work

Vocabulary extension Most state-of-the-art NLP models are deep neural networks with a vocabulary embedding layer as their first layer. For ease of understanding, we can imagine the embedding layer as a table with thousands of rows. Each row corresponds to an existing word, sub-word [19], letter, or punctuation sign in a natural language, such as English. This embedding layer works along with a tokeniser, which transforms the string prompt into a sequence of tokens, corresponding to the row indices of the embedding table. We refer to one row of this embedding table as the embedding of a token. One embedding is a continuous vector representation allowing processing by further layers of the text encoder. Existing research [5, 10, 20] extends the embedding table for specific tasks. For example, Tai *et al* [20] extend the vocabulary embedding to account for words that are specific to the biomedical domain. In our approach, detailed in Section 3, we extend the embedding table by one embedding, *i.e.* one row, and tune this embedding to match a text description given as input.

Text-to-image models and their text encoders Text-to-image models generate images from natural language descriptions, referred to as prompts. Recent text-to-image models [11, 17, 18] pre-process the prompt given as input with a text encoder. The encoded prompt is then used to generate the image from, as shown in Figure 2. State-of-the-art models typically use a CLIP [13] or a T5 [14] model as text encoder. In this paper, we implemented our method for Stable Diffusion [17], which uses a CLIP ViT L/14 text encoder.

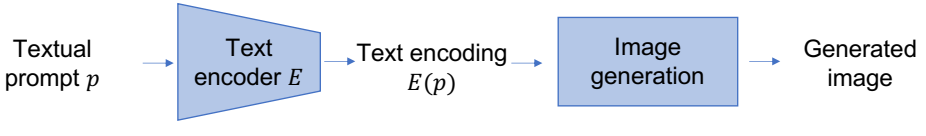

Figure 2: **Overall architecture of typical large-scale text-to-image models.** Text-to-image models typically use a text encoder and an image-generation module. The text encoding, obtained from the prompt, is used as input to the image-generation part. In the case of Stable Diffusion, the image generation consists of using a text-conditioned U-Net to iteratively denoise latent codes, and a latent decoder to decode the final latent code into an image.

Expanding the vocabulary of text-to-image models Vocabulary expansion of text-to-image models has practical applications such as improving character consistency in story visualisation [12]. Given sample images of an object, Textual Inversion [4] and Custom Diffusion [9] extend the vocabulary of the text-to-image model with a new token, allowing users to generate novel images of a specific concept.

Textual Inversion optimises the embedding of this token with the original diffusion loss, *i.e.* such that the generated images contain the same visual features as the sample images. In addition, Custom Diffusion also tunes some parameters in the image-generation part. The optimisation process is time-consuming for both methods. Textual inversion, for instance, takes roughly one hour per optimisation on a V100 GPU [7], as optimisation relies on supervision with the original diffusion loss. Apart from the high computational cost and need for sample images, replicating visual details from existing images may also diminish the element of surprise or novelty that one expects from text-to-image model outputs.

Conversely, our approach VETIM can embed concepts in the vocabulary of a text-to-image model without sample images. The generated images do not mimic any sample images, and VETIM does not need to access image-generation to optimise the embedding, making it more efficient and faster: only around 2 minutes on a V100 GPU.

3 VETIM

In this section, we describe our approach in more detail. VETIM is a technique for inverting a text description into a vocabulary embedding of a text-to-image model. To optimise the embedding, VETIM uses supervision solely at the text encoder output.

Method and architecture Our method, VETIM, takes as input a reference description of a concept, *e.g.* an object or a character of a story. Our objective is to learn an embedding of this concept in the vocabulary of the model. The embedding can then directly be used to compose new textual prompts. For training, we modify the tokeniser and embedding layer of the text encoder to allow processing with a new token. As in Textual Inversion [4], we refer to this token as S_*. We add a new row v_* in the vocabulary embedding layer of the text encoder, and modify the tokeniser accordingly so that S_* is correctly embedded into v_*. The remaining elements of the text encoder are frozen, and, importantly, the image-generation part of the text-to-image model is not accessed at all during optimisation.

Initialisation of the embedding v_* Before starting the optimisation, we initialise the embedding v_* with the values of another embedding that already exists in the text-encoder vocabulary. To do this, for all existing tokens w, we compute the cosine similarity between the text encodings of "A photo of w" and of "A photo of t", where t refers to the input description, *e.g.* "a small, brilliant red stone that can produce the Elixir of Life and turn base metals into gold". We then initialise v_* with the embedding of the token w with the highest computed similarity. As an example, with the CLIP ViT L/14 text encoder, the token *blueberries* is the most similar token to the description "a furry, midnight blue rectangle with small, glowing dots on its surface".

Optimisation Our optimisation objective is to match the token S_* to a concept described by a reference description t, such that textual prompts using S_* or the description t generate

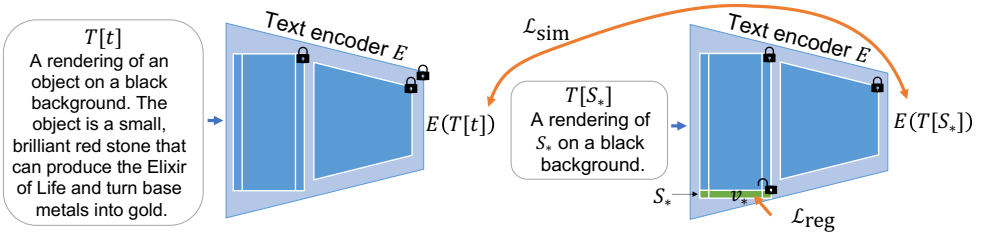

Figure 3: **Optimisation.** Given a reference description t of a concept, we optimise the vocabulary embedding v_* of the new token S_* by maximising the similarity between the encodings of $T[S_*]$ and $T[t]$. These encodings are obtained with a frozen text encoder. The only optimised parameters are those of the new vocabulary embedding v_*. A regularisation term is used to stabilise the norm of v_*, keeping it similar to other vocabulary embeddings.

similar images. To achieve this goal, we only train the new embedding vector v_*. Note that, since only v_* is optimised, the encodings of the prompts that do not use S_* remain unchanged, preventing any catastrophic forgetting.

Each training sample in our optimisation loop consists of two texts $T[S_*]$ and $T[t]$ using a template $T[.]$. The text $T[S_*]$ contains the token S_*, for instance, $T[S_*]$ ="A rendering of S_* on a black background." The text $T[t]$ is similar, but contains the reference description t, for instance, $T[t]$ ="A rendering of an object on a black background. The object is a small, brilliant red stone that can produce the Elixir of Life and turn base metals into gold." Since we want these two texts to generate similar images, we optimise the embedding v_* to maximise the cosine-similarity of the two encodings $E(T[S_*])$ and $E(T[t])$ obtained via the text-encoder E. For each training sample, the template $T[.]$ is randomly sampled from a list. Details are provided in the supplementary material. Figure 3 illustrates our procedure.

Our loss is composed of two terms $\mathcal{L} = \mathcal{L}_{\text{sim}} + \alpha\mathcal{L}_{\text{reg}}$. The term $\mathcal{L}_{\text{sim}}$ aligns the encodings of the two texts $T[S_*]$ and $T[t]$ by maximising their cosine-similarity, as explained above. So, $\mathcal{L}_{\text{sim}} = 1 - \cos(E(T[S_*]), E(T[t]))$. Since we have no supervision from image-generation, a regularisation term is important to keep the embedding v_* close to other embeddings, and to allow the image-generation part to generalise properly to the new embedding. A term $\mathcal{L}_{\text{reg}}$ therefore regularises the norm of v_* towards the average norm of embeddings of all other tokens. So, $\mathcal{L}_{\text{reg}} = \left(\|v_*\| - \mathbb{E}_{v\in\text{embeddings}}(\|v\|)\right)^2$.

Implementation details We implemented our approach VETIM using Stable Diffusion v1.5 [17], which uses the text encoder from CLIP ViT L/14 [13]. However, VETIM could be similarly implemented in other text-to-image models. When using Stable Diffusion, the text-encoding $E(p)$ of a prompt p is represented by a set of vectors, each vector corresponding to a token in the prompt. In order to compute the loss term $\mathcal{L}_{\text{sim}}$, it is necessary to pool the text-encoding $E(p)$, as was done in the original training of CLIP [13], to obtain a single vector that represents the entire prompt. For each concept, we perform 50 optimisation steps with a batch size of 512 samples and an effective learning rate of 10^{-6}. We use Adam optimiser [8] with $\alpha = 100$. At the end of the optimisation, we may, depending on the application, overwrite an existing embedding with our optimised one or retain it as an additional embedding. This makes our approach flexible and extensible.

Computational budget VETIM optimises only a few hundred parameters, the size corresponding to a single vocabulary embedding. The optimisation also only involves the text encoder, making it significantly faster than, for instance, generating images from the text description and then using Textual Inversion [4]. In the case of Stable Diffusion v1.5, we only optimise 768 parameters, the dimensionality of v_*. On a V100 GPU, our optimisation takes around 2 minutes, only as much time as required for generating approximately 30 images. In comparison, Textual Inversion takes around 1 hour per concept on the same GPU [7].

4 Evaluation

To test and evaluate our approach, we gathered an evaluation dataset with descriptions of 150 objects, fictitious and otherwise, provided in the supplementary material. We then use VETIM to expand the vocabulary of Stable Diffusion with new tokens for these objects. We show that using our new customised tokens produces similar images as when using lengthy input descriptions.

4.1 Qualitative evaluation

Figures 1 and 4 show that our approach VETIM is able to learn the concepts from the input descriptions, and compose them into new prompts. The rows labelled *VETIM optimised embedding* in Figure 4, similarly as the bottom row of Figure 1, contain images generated with our token S_*, with the four prompts "A photo of S_*", "A photo of S_* on the Moon", "An oil painting of S_*", and "Elmo holding S_*". These generated images are indeed similar to the ones of the rows labelled *Description (groundtruth)* which, similarly as the top row of Figure 1, contain the images generated from the input description, *e.g.* with the prompt "A photo of an object on the Moon. The object is a twisted, abstract sculpture made of delicate, interlocking tendrils of glass." We aim to have similar images for these two rows. Additionally, we show, in the rows labelled *VETIM initialisation embedding* of Figure 4, the images generated using the existing token that is the most similar to the description.

4.2 Quantitative evaluation

To evaluate our approach, we assess the quality of vocabulary embeddings along two properties, **reconstruction** and **composability**. Gal *et al.* [4] and Kumari *et al.* [9] use resembling metrics, referring to reconstruction as image-alignment, and composability as text-alignment or editability.

Reconstruction score: Our reconstruction score measures how well the vocabulary embedding can reconstruct the concept. It is computed by generating 16 images from the prompt "A photo of S_*" and computing their average CLIP similarity with the same prompt containing the input description instead of S_*. We chose the number 16 arbitrarily.

Composability score: Our composability score estimates how well the embedding can be used inside other prompts. It is computed by generating 16 images for each of the three prompts "A photo of S_* on the Moon", "An oil painting of S_*", and "Elmo holding S_*", and computing their average CLIP similarity with the same prompts containing the tokens *an object*, in place of S_*. This composability score then indicates how well the generated images contain/compose the elements in these prompts, disregarding the reconstruction of the concept itself.

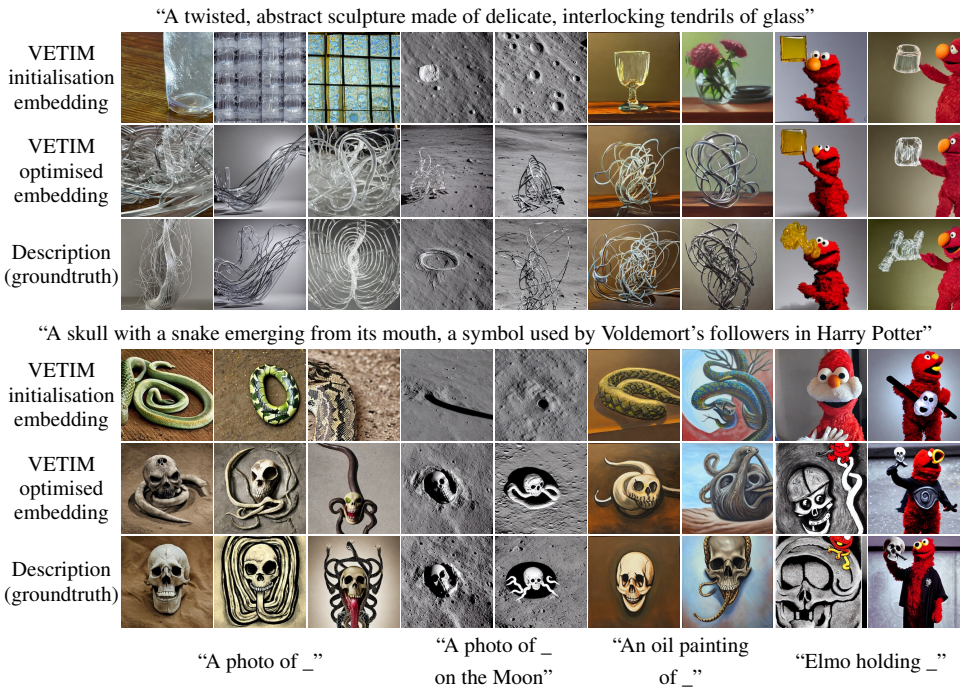

Figure 4: **Qualitative text-to-image results.** The rows *VETIM initialisation embedding* and *VETIM optimised embedding* contain images generated with the prompts "A photo of S_*", "A photo of S_* on the Moon", "An oil painting of S_*", and "Elmo holding S_*". For the first row, *VETIM initialisation embedding*, the embedding of S_* was replaced by the embedding of the token that is the closest to the description by cosine-similarity, here, the tokens *glass* (first description) and *ssss* (second description). For the second row, *VETIM optimised embedding*, the embedding of S_* corresponds to the embedding optimised with our method VETIM. The bottom row, *Description*, contains images generated with the input text description, *e.g.* "Elmo holding an object. The object is a skull with a snake emerging from its mouth, a symbol used by Voldemort's followers in Harry Potter." for the two rightmost images of the bottom row. For each of the two given descriptions, the same seed was used to generate images in a given column. See the supplementary material for additional qualitative results.

Reconstruction-composability results The average value of those two metrics, computed over the 150 object descriptions of our dataset, is reported in Figure 5 in a 2D reconstruction-composability plot. Numerical values with standard deviations and details are reported in the supplementary material.

In addition to the setups considered in the qualitative evaluation section, namely *VETIM initialisation embedding*, *VETIM optimised embedding*, and *Description*, we also evaluate the embedding obtained by averaging the embeddings of every token of the text description and normalising it to the average norm of other embeddings. This embedding is labelled as *Avg. description* in Figure 5. While it intuitively seems a reasonable method for shortening a description down to a single token, this method leads to the worst reconstruction.

Avg. description is an example of a text-based vocabulary expansion method that leads

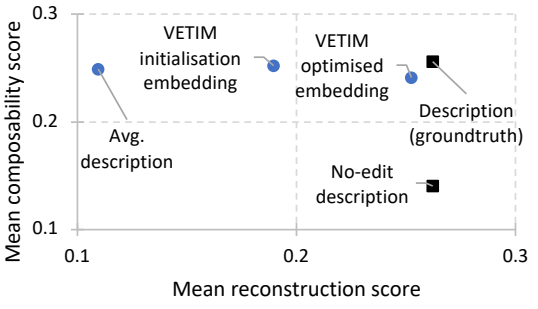

Figure 5: **Reconstruction *vs.* composability plot.** This plot compares our method with others for the task of learning an embedding from a text description (blue circles). To provide an intuitive understanding of the scales, we also show reference textual descriptions which are not embeddings (black squares).

to embeddings with low reconstruction and high composability. To provide an intuition for the scale of the composability score, we also computed a reference low composability score. That is, we compute the score of an embedding that allows reconstructing the concept, but which cannot be composed with different prompts. To compute this reference composability score, we generate the images from the input descriptions, but always using the template "A photo of _" instead of the prompts "A photo of _ on the Moon", "An oil painting of _", and "Elmo holding _". This provides a score reference for low composability and high reconstruction, and is labelled *No-edit description* in Figure 5.

Overall, our approach is better than any other approach in terms of reconstruction, yet it has lower composability. This is not surprising however, as prior research [4] shows a trade-off between reconstruction and composability of embeddings.

5 Applications

In addition to the practicality of having custom tokens in text-to-image models, VETIM enables, by condensing a text description into a single token, to use text descriptions in techniques that exclusively operate at the token level. These techniques include, overwriting biased embeddings, interpreting attention maps, and swapping one token with another. Additional details on the results are also provided in the supplementary material.

Note that these applications could also be done using the embedding learned with Textual Inversion [4]. Our approach VETIM achieves these applications without requiring any sample images of each concept, but only textual descriptions.

Debiasing the vocabulary of a model Bansal *et al.* [1] show that images generated by Stable Diffusion with debiased expressions like "firefighter, if all genders can be a firefighter" are less gender-biased than the ones generated with "firefighter" directly. However, requesting each user of Stable Diffusion to debias their prompts is impractical. To prevent any access to the original biased embedding, we need to overwrite it with a new unbiased embedding, similarly as Gal *et al.* [4] do for this application. For instance, we overwrite the biased embedding of the token *firefighter* with the one optimised by VETIM from the debiased expression "firefighter, if all genders can be a firefighter". We performed this experiment, and obtained a customised version of Stable Diffusion generating less biased images of professions, as we show in Figure 6.

Figure 6: **Debiasing *firefighter* and *doctor* with VETIM.** Top row: Images generated with original Stable Diffusion, Bottom row: Images generated with our customised Stable Diffusion, obtained by overwriting the embeddings of the tokens *firefighter* and *doctor* by those learned by VETIM from debiased expressions. Images are generated from the prompts "A photo of a firefighter" (left) and "A photo of a doctor" (right). While the original Stable Diffusion model generates almost exclusively masculine attributes for these professions, our customised Stable Diffusion model generates more gender-varied images.

Improving interpretability with cross-attention maps Images generated with Stable Diffusion can be interpreted using diffusion attentive attribution maps (DAAM) [21]. For each token, a DAAM is obtained by aggregating the attention maps of the cross-attention layers [22] of the U-Net. However, descriptions of objects are still hard to interpret, because the DAAMs of several tokens need to be considered at once. As illustrated in Figure 7, images generated with tokens from VETIM are easier to interpret.

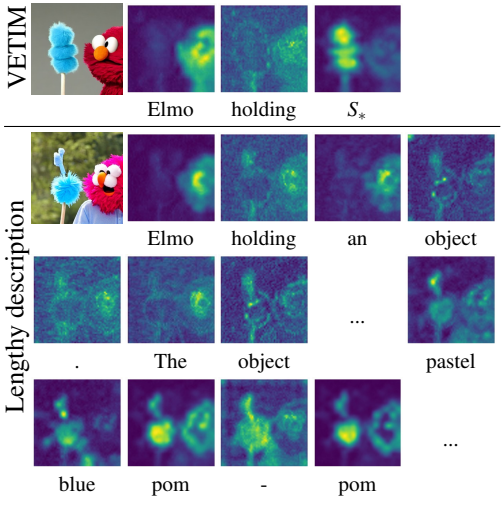

Figure 7: **Diffusion attentive attribution maps.** We used VETIM to learn a token S_* from the description "a fluffy, pastel blue pom-pom attached to a long, thin stick". The top image was generated from the prompt "Elmo holding S_*", and the bottom image was generated from the lengthy prompt "Elmo holding an object. The object is a fluffy, pastel blue pom-pom attached to a long, thin stick." While the two generated images are similar, it is easier to interpret the generation at the top. Especially, while there are many DAAMs [21] to consider at the bottom, the DAAM for the token S_* at the top clearly indicates which pixels correspond to the object.

Word-swap with descriptions To edit images generated by Stable Diffusion, Hertz *et al.* [6] introduce a method, referred to as word-swap or `AttentionReplace`, which enables to change one word of the textual prompt while keeping the remaining elements of the generated images unchanged. In short, word-swap injects the attention maps of the initial image generation into the generation of the modified prompt. Word-swap (`AttentionReplace`) only works at the token level and cannot be used as it is to replace a word with a lengthy

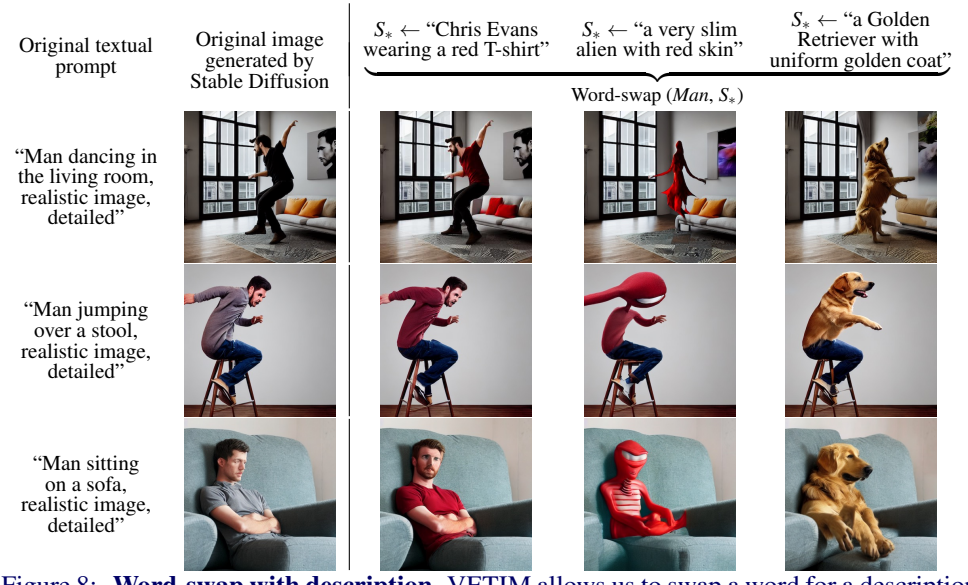

Figure 8: **Word-swap with description.** VETIM allows us to swap a word for a description while maintaining the functionality of the original word-swap method [6]. The images in the second column are generated from the prompts in the first column. The remaining images are generated with word-swap, with tokens learned by VETIM from the descriptions.

description. However, using VETIM, we can learn a new word from the lengthy description, and then use the word-swap approach. This enables finer control over the generated images. By using VETIM and word-swap, it is possible to obtain the images of Figure 8.

6 Conclusion and future work

We present VETIM, an intuitive approach that optimises new vocabulary embeddings in Stable Diffusion using supervision solely at the text encoder output. This contrasts with previous research, which uses the original diffusion loss to optimise new vocabulary embeddings. Our optimisation is simpler and significantly faster, taking only the time of a few tens of inferences with Stable Diffusion. Interestingly, Stable Diffusion is able to generalise to embeddings optimised without involving the image-generation part.

Instead of learning to replicate visual features from sample images, our method VETIM learns a new vocabulary embedding that generates similar images as from an input text description, bypassing the need for the sample images to expand the vocabulary.

Future research may consider integrating VETIM as an extra supervision term in Textual-Inversion-like methods, potentially benefiting from the faster convergence of VETIM with the extra capability of Textual Inversion, *e.g.* learning new visual features not contained in the original model.

Acknowledgements

This work was supported by Innosuisse grant 48552.1 IP-ICT. The authors thank Athanasios Fitsios and the members of the Image and Visual Representation Lab (IVRL) for their advice.

References

[1] Hritik Bansal, Da Yin, Masoud Monajatipoor, and Kai-Wei Chang. How well can Text-to-Image Generative Models understand Ethical Natural Language Interventions? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1358–1370, 2022.

[2] Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. VQGAN-CLIP: Open domain image generation and editing with natural language guidance. In *European Conference on Computer Vision*, pages 88–105. Springer, 2022.

[3] Boris Dayma, Suraj Patil, Pedro Cuenca, Khalid Saifullah, Tanishq Abraham, Phúc Lê Khac, Luke Melas, and Ritobrata Ghosh. DALL·E Mini, 7 2021. URL https://github.com/borisdayma/dalle-mini.

[4] 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. In *The Eleventh International Conference on Learning Representations*, 2022.

[5] Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. WARP: Word-level Adversarial ReProgramming. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4921–4933, 2021.

[6] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-Prompt Image Editing with Cross Attention Control. In *The Eleventh International Conference on Learning Representations*, 2022.

[7] HuggingFace. Textual Inversion — huggingface.co. https://huggingface.co/docs/diffusers/training/text_inversion, 2022.

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

[9] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-Concept Customization of Text-to-Image Diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1931–1941, 2023.

[10] Brian Lester, Rami Al-Rfou, and Noah Constant. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, 2021.

[11] Midjourney. Midjourney, 2022. URL https://www.midjourney.com/.

[12] Xichen Pan, Pengda Qin, Yuhong Li, Hui Xue, and Wenhu Chen. Synthesizing Coherent Story with Auto-Regressive Latent Diffusion Models. *arXiv preprint arXiv:2211.10950*, 2022.

[13] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[14] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.

[15] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[16] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022.

[17] 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.

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

[19] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, 2016.

[20] Wen Tai, HT Kung, Xin Luna Dong, Marcus Comiter, and Chang-Fu Kuo. exBERT: Extending pre-trained models with domain-specific vocabulary under constrained training resources. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1433–1439, 2020.

[21] Raphael Tang, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Jimmy Lin, and Ferhan Ture. What the DAAM: Interpreting Stable Diffusion Using Cross Attention. *arXiv preprint arXiv:2210.04885*, 2022.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.