Supplementary Material 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

In this document, we provide supplementary material that additionally supports the claims of the manuscript. This document is structured as follows:

- Appendix A provides details on our optimisation technique.
- Appendix B presents information regarding our evaluation dataset, which comprises 150 object descriptions.
- Appendix C provides a more exhaustive quantitative evaluation.
- Appendix D contains supplementary qualitative results.
- Appendix E features additional results related to the applications described in the main paper.
- Appendix F discusses the limitations of our approach.

A Implementation details: training samples

We expand here the description of our optimisation strategy, described in Section 3 of the manuscript.

Textual Inversion [1] The main source of inspiration for our work was Textual Inversion [1]. Textual Inversion proposes to learn the embedding of a new token S_*, for a concept described by sample images. The embedding v_* of S_* is optimised with the original diffusion loss, *i.e.* such that a noisy version of a sample image is correctly denoised into the sample image. When textual prompts contain the token $S*$, the model then generates images that mimic the visual features learned from the sample images.

Each training sample in Textual Inversion is a text-image pair composed of a text $T[S_*]$ and a sample image I. The text $T[S_*]$ is a generic prompt containing the token S_* , *e.g.* "a close-up photo of a S_*". To generate $T[S_*]$, a template $T[.]$ is sampled from a predefined list, for instance, one of these in Tables 1 and 2. The authors of Textual Inversion noted that using directly S_* instead of $T[S_*]$ gives consistently worse results. Apart from the text $T[S_*]$, an image I is selected randomly from the input sample images.

The training signal in Textual Inversion is as follows. The VAE encoding $\mathcal{E}(I)$ of the image I is noised, according to a predefined noise schedule and a random time-step. The U-Net of Stable Diffusion, trained to predict the noise from the noisy version of $\mathcal{E}(I)$, outputs a predicted noise. A L2 loss on the noise reconstruction, *i.e.* the diffusion loss, is used as training signal, and back-propagated to the embedding v_*.

This optimisation of the embedding v_* in Textual Inversion involves back-propagating the training signal through the image-generation part (for Stable Diffusion, through the U-Net) and through the text encoder, both of them being frozen. The overall optimisation is relatively lengthy, around 1 hour on a V100 GPU [2], making it worthwhile to look for potential auxiliary supervision that could speed up the convergence of the embedding v_*. Our manuscript presents such an alternative supervision, at the output of the text encoder. Our supervision alone does not need any forward or backward pass through the image-generation part of the model, and does not involve noising images. Our optimisation is then significantly faster and simpler.

From images to textual description In our manuscript, we consider the text-only setting, in which we do not have sample images as input, but only a textual description t.

Instead of being text-image pairs $(T[S_*], I)$ as in Textual Inversion, our training samples are then pairs composed of two text strings, one text containing the token S_*, the other containing the textual description t. As we aim to match the token S_* with the description t, the only change between the two texts of the pair should be S_* and t. Hence, our training samples are pairs of texts $(T[S_*], T[t])$, where the token S_* and the description t are put in the same, randomly selected, template $T[.]$.

For instance, with the template $T[.]$ ="a photo of a _", we could have:

- $T[S_*]$ = "a photo of a S_*"
- $T[t]$ = "a photo of a small, brilliant red stone that can produce the Elixir of Life and turn base metals into gold"

- $T[S_*]$ = "a photo of a S_*"
- $T[t]$ = "a photo of a elongated curved tropical fruit of a plant, which grows in bunches and has a creamy flesh and a smooth skin"

Complex templates During our preliminary experiments, we found that generic templates do not lead to good composability of the optimised embedding. In order to improve composability, *i.e.* for the embedding to "work well" in more complex prompts, we optimise it within complex prompts as well. We want to use more complex templates like $T[.]$ ="A rendering of _ on a black background." For simplicity and to keep a grammatically correct text for $T[t]$, we append the description t at the end of the templates $T[.]$, and use a class of the object (typically the words "an object") in place of the underscore.

For instance, with the template $T[.]$ ="A rendering of _ on a black background.", we now have:

a photo of a _
a rendering of a _
a cropped photo of the _
the photo of a _
a photo of a clean _
a photo of a dirty _
a dark photo of the _
a photo of my _
a photo of the cool _
a close-up photo of a _
a bright photo of the _
a cropped photo of a _
a photo of the _
a good photo of the _
a photo of one _
a close-up photo of the _
a rendition of the _
a photo of the clean _
a rendition of a _
a photo of a nice _
a good photo of a _
a photo of the nice _
a photo of the small _
a photo of the weird _
a photo of the large _
a photo of a cool _
a photo of a small _

Table 1: List of templates $T[.]$ from Textual Inversion [11]

a photo of a _	a rendering of a _
a cropped photo of the _	the photo of a _
a photo of a clean _	a photo of a dirty _
a dark photo of the _	a photo of my _
a photo of the cool _	a close-up photo of a _
a bright photo of the _	a cropped photo of a _
a photo of the _	a good photo of the _
a photo of one _	a close-up photo of the _
a rendition of the _	a photo of the clean _
a rendition of a _	a photo of a nice _
a good photo of a _	a photo of the nice _
a photo of the small _	a photo of the weird _
a photo of the large _	a photo of a cool _
a photo of a small _	an illustration of a _
a rendering of a _	a cropped photo of the _
the photo of a _	an illustration of a clean _
an illustration of a dirty _	a dark photo of the _
an illustration of my _	an illustration of the cool _
a close-up photo of a _	a bright photo of the _
a cropped photo of a _	an illustration of the _
a good photo of the _	an illustration of one _
a close-up photo of the _	a rendition of the _
an illustration of the clean _	a rendition of a _
an illustration of a nice _	a good photo of a _
an illustration of the nice _	an illustration of the small _
an illustration of the weird _	an illustration of the large _
an illustration of a cool _	an illustration of a small _
a depiction of a _	a rendering of a _
a cropped photo of the _	the photo of a _
a depiction of a clean _	a depiction of a dirty _
a dark photo of the _	a depiction of my _
a depiction of the cool _	a close-up photo of a _
a bright photo of the _	a cropped photo of a _
a depiction of the _	a good photo of the _
a depiction of one _	a close-up photo of the _
a rendition of the _	a depiction of the clean _
a rendition of a _	a depiction of a nice _
a good photo of a _	a depiction of the nice _
a depiction of the small _	a depiction of the weird _
a depiction of the large _	a depiction of a cool _
a depiction of a small _	

Table 2: Extended list of templates $T[.]$ from Textual Inversion [11]

- $T[S_*]$ = "A rendering of S_* on a black background."
- $T[t]$ = "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."

- $T[S_*]$ = "A rendering of S_* on a black background."
- $T[t]$ = "A rendering of a fruit on a black background. The fruit is an elongated curved tropical fruit of a plant, which grows in bunches and has a creamy flesh and a smooth skin."

The list of complex templates can be generated from different components, for instance, styles, backgrounds, and other objects. All experiments in the main document, except debiasing (see Appendix E.1), use the following templates. A `styles` variable contains a list of different styles that can be applied to the description, such as "a photo of", "a rendering of", "a painting of", and so on. A `backgrounds` variable holds a list of various backgrounds on which the image can be situated, such as "in the sky", "on a black background", "on a white background", and so on. An `other_objects` variable consists of a list of different objects or subjects that can be included in the image description, such as "a train", "a boy", and so on. We provide the lists we use in Tables 3, 4, and 5. A loop then creates complex templates by combining these components in various ways. It generates templates in the following formats:

- "`style` _ `background`." - Combining a style, the concept, and a background, for instance "a photo of _ in the sky."
- "`style` _ and `other_object`." - Combining a style, the concept, and another object, for instance "a photo of _ and a train."
- "_ `background` with `other_object`." - Combining the concept, a background, and another object, for instance "_ in the sky with a train."
- "`other_object` `background` with _." - Combining another object, a background, and the concept, for instance "a train in the sky with _."
- "_ and `other_object` `background`." - Combining the concept, another object, and a background, for instance "_ and a train in the sky."
- "`other_object` and _ `background`." - Combining another object, the concept, and a background, for instance "a train and _ in the sky."

The loop iterates through the `styles`, `backgrounds`, and `other_objects`, creating a template for each combination. Note that to avoid train/evaluation leakage, we explicitly did not train with the templates used for composability evaluation. Especially, "oil painting", "on the Moon", and "Elmo holding" are not seen during the optimisation of the embedding.

In summary, to generate a training sample for VETIM, a template $T[.]$ is randomly selected from a predefined list generated by combining components. The token S_* and the input description t are put in the template $T[.]$, forming a training sample $(T[S_*], T[t])$.

B Evaluation dataset

As mentioned in Section 4 of the manuscript, we gathered a dataset of 150 object descriptions to evaluate our method. To fit the templates described in Appendix A, each object description t contains the following elements:

a photo of
a rendering of
a rendition of
this is
a painting of
charcoal drawing of

Table 3: List of different styles used to create the templates $T[.]$

in the sky
on a black background
on a white background
on a green background
on a blue background
on a red background
at work
in the street
in the bus

Table 4: List of various backgrounds used to create the templates $T[.]$

a boy
family
the Santa Claus
a train
a car
a table
a house

Table 5: List of objects/subjects used to create the templates $T[.]$

- A noun-phrase description of the concept being referred to. It describes the visual characteristics or features of the concept in a general manner. For example, "a small, brilliant red stone that can produce the Elixir of Life and turn base metals into gold" or "an elongated curved tropical fruit of a plant, which grows in bunches and has a creamy flesh and a smooth skin"
- The class or category to which the concept belongs. It provides a way to refer to the concept. For example, "an object" or "a fruit".
- A rephrasing of two previous points into a grammatically correct sentence. For example, "The object is a small, brilliant red stone that can produce the Elixir of Life and turn base metals into gold." or "The fruit is an elongated curved tropical fruit of a plant, which grows in bunches and has a creamy flesh and a smooth skin."

To cover more diverse cases and to ease future work, the 150 object descriptions can be categorised into 3 types:

- 47 out of the 150 objects are real-world existing objects, for which images and descriptions can easily be found, including for instance "banana", "screwdriver", "measuring tape", etc. For these 47 objects, we gathered a description from `en.wiktionary.org`, with a few manual changes. We selected the 47 objects to correspond to the 47 object classes of the GoLD dataset [3], to allow future work to be compared across different modalities. To ensure the initialisation embedding (see Section 3, Paragraph *Initialisation of the embedding* v_* of the manuscript) is not the one of a token already referring to the object (*e.g.* the tokens *banana* or *bananas*), we only consider tokens with a Levenshtein distance above 2.
- 26 out of the 150 objects correspond to visual descriptions of objects from famous fiction stories. We generated these visual descriptions using ChatGPT ([2], March 23rd version), from the prompt "Generate short visual descriptions for many objects from famous books. Answer with a JSON dictionary. Include objects from various stories and keep descriptions very short."
- The 77 remaining objects correspond to visual descriptions of objects that do not exist. We generated these visual descriptions using ChatGPT ([4], March 23rd and May 3rd versions), from the prompt "Generate a list of short visual descriptions of objects that do not already usually exist in a dictionary. Focus your descriptions on the visual appearance of the objects, not their functionality."

We refer to these 3 subsets as (respectively) *existing objects*, *fiction stories objects*, and *not-*

existing objects. The 150 object descriptions are provided as a `csv` file in the supplementary material.

C Details on quantitative evaluation

We provide below a more detailed quantitative evaluation of our approach.

C.1 Evaluation on the 150 object descriptions

Table 6 contains numerical values associated with Figure 5 of the manuscript.

Reconstruction score Recall that the reconstruction score measures how well the vocabulary embedding can reconstruct a concept. To compute this score for one embedding, 16 images are generated from the prompt "A photo of S_*" (where S_* represents the token associated to the embedding). The average CLIP similarity is then calculated between these 16 generated images and the same prompt containing the input description instead of "A photo of t", where t is the input description (noun-phrase). The reconstruction score refers to this average of 16 CLIP similarity scores. The process is repeated for all 150 objects in the evaluation set, and the mean of the reconstruction scores is obtained.

Composability score Recall that the composability score evaluates how well the embedding can be used in different prompts. The composability score is computed by generating 48 images in total: 16 images for each of the three prompts - "A photo of S_* on the Moon", "An oil painting of S_*", and "Elmo holding S_*". Here, S_* represents the token associated with the embedding being evaluated. Once the 48 images are generated, the average CLIP similarity is computed between these generated images and their respective modified prompts - "A photo of an object on the Moon", "An oil painting of an object", and "Elmo holding an object". The composability score refers to this average of 48 CLIP similarity scores. This composability score indicates how well the generated images capture or compose the elements described in the prompts, regardless of the accuracy of S_* reconstructing the concept itself. Again, the process is repeated for all 150 objects in the evaluation set, and the mean of the composability scores is obtained.

The low standard errors of the mean reconstruction score and mean composability score shown in Table 6 indicate that our evaluation is reliable, because the difference in scores between any pair of methods is higher than the standard errors.

Method	Reconstruction scores			Composability scores		
	Mean	Standard error of the mean	Standard deviation	Mean	Standard error of the mean	Standard deviation
Avg. description	0.109	0.002	0.024	<u>0.2491</u>	0.0003	0.0040
VETIM initialisation embedding	<u>0.190</u>	0.002	0.027	**<u>0.2523</u>**	0.0009	0.0115
VETIM optimised embedding	**<u>0.252</u>**	0.002	0.028	0.241	0.002	0.022
Description (groundtruth)	0.262	0.003	0.031	0.256	0.001	0.017
No-edit description				0.140	0.001	0.017

Table 6: **Numerical values of the quantitative evaluation.** Best scores among the 3 evaluated methods are **<u>bold and underlined</u>**. Second best scores among the 3 evaluated methods are <u>underlined</u>. The scores for the two reference settings *Description* and *No-edit description* are also provided.

C.2 Details on the composability scores

The **composability score** of an embedding, as mentioned earlier, is the average CLIP similarity score between the 48 generated images and their respective modified prompts (replacing S_* by "an object") for all three prompts: "A photo of S_* on the Moon", "An oil painting of S_*", and "Elmo holding S_*". It represents an overall assessment of how well the embedding can be used in different prompts. This composability score can be split into 3 values: **background composability score**, **style composability score**, and **object composability score**. They focus on the individual prompts and calculate the average of the 16 CLIP similarity scores for each prompt separately. For example, the **background composability score** of an embedding is obtained by calculating the average CLIP similarity between the 16 images generated from the prompt "A photo of S_* on the Moon" and the modified prompt "A photo of an object on the Moon". This score specifically measures the embedding's ability to compose in prompts with background change, evaluated here by changing the background to the Moon.

The composability score provides an overall evaluation of the embedding's performance across all three prompts, while the background composability score, style composability score, and object composability score offer specific assessments for each individual prompt. Table 7 reports the average of the 150 background composability scores, of the 150 style composability scores, and of the 150 object composability scores.

As seen in Tables 6 and 7, our optimised embeddings with VETIM have better reconstruction, yet they have lower background and object composability scores.

Method	Style composability scores		Background composability scores		Object composability scores	
	Mean	Standard error	Mean	Standard error	Mean	Standard error
Avg. description	0.2288	0.0006	0.2417	0.0004	**0.2767**	0.0003
VETIM initialisation embedding	0.243	0.001	**0.246**	0.001	0.269	0.001
VETIM optimised embedding	**0.250**	0.002	0.225	0.003	0.249	0.003
Description (groundtruth)	0.248	0.002	0.251	0.002	0.270	0.002
No-edit description	0.166	0.002	0.144	0.002	0.112	0.002

Table 7: **Detailed numerical values of the quantitative evaluation for the composability scores.** Best scores among the 3 evaluated methods are **bold and underlined**. Second best scores among the 3 evaluated methods are underlined. The scores for the two reference settings *Description* and *No-edit description* are also provided.

C.3 Evaluation on the 3 subsets

We further conduct the quantitative evaluation for each of the three subsets of the evaluation dataset described in Appendix B. The results are provided in Tables 8, 9, and 10. Observations made on our whole evaluation set remain valid for the three subsets individually.

Method	Reconstruction scores		Composability scores	
	Mean	Standard error of the mean	Mean	Standard error of the mean
Avg. description	0.121	0.004	0.2498	0.0007
VETIM initialisation embedding	0.201	0.004	**0.257**	0.001
VETIM optimised embedding	**0.239**	0.003	0.242	0.004
Description (groundtruth)	0.238	0.003	0.264	0.002
No-edit description			0.128	0.002

Table 8: **Numerical values of the quantitative evaluation on the *existing objects* subset.**

Method	Reconstruction scores		Composability scores	
	Mean	Standard error of the mean	Mean	Standard error of the mean
Avg. description	0.112	0.004	0.2477	0.0004
VETIM initialisation embedding	0.191	0.004	**0.249**	0.003
VETIM optimised embedding	**0.254**	0.006	0.238	0.005
Description (groundtruth)	0.271	0.006	0.249	0.005
No-edit description			0.138	0.003

Table 9: **Numerical values of the quantitative evaluation on the *fiction stories objects* subset.**

Method	Reconstruction scores		Composability scores	
	Mean	Standard error of the mean	Mean	Standard error of the mean
Avg. description	0.102	0.002	0.2491	0.0004
VETIM initialisation embedding	0.182	0.003	**0.251**	0.001
VETIM optimised embedding	**0.260**	0.003	0.242	0.002
Description (groundtruth)	0.273	0.003	0.253	0.001
No-edit description			0.149	0.001

Table 10: **Numerical values of the quantitative evaluation on the *not-existing objects* subset.** Best scores among the 3 evaluated methods are **bold and underlined**. Second best scores among the 3 evaluated methods are underlined. The scores for the two reference settings *Description* and *No-edit description* are also provided.

D Additional qualitative results

We provide additional qualitative results of our approach in Figures 9 to 17. Similarly as Figure 4 of the manuscript, we generate images from the 4 prompts "A photo of S_*", "A photo of S_* on the Moon", "An oil painting of S_*" and "Elmo holding S_*", where the token S_* is either associated to the initialisation embedding of our method VETIM (first rows in Figures 9 to 17), or to the embedding optimised with our method VETIM (second rows in

Figures 9 to 17). We also generate images from the input descriptions directly (third rows in Figures 9 to 17), without shortening it into a single token. This is similar to Figure 4 of the manuscript.

Again, we can see that our optimised embedding learned with VETIM is as effective as generating images from the input textual description, with the additional benefit of using a single token.

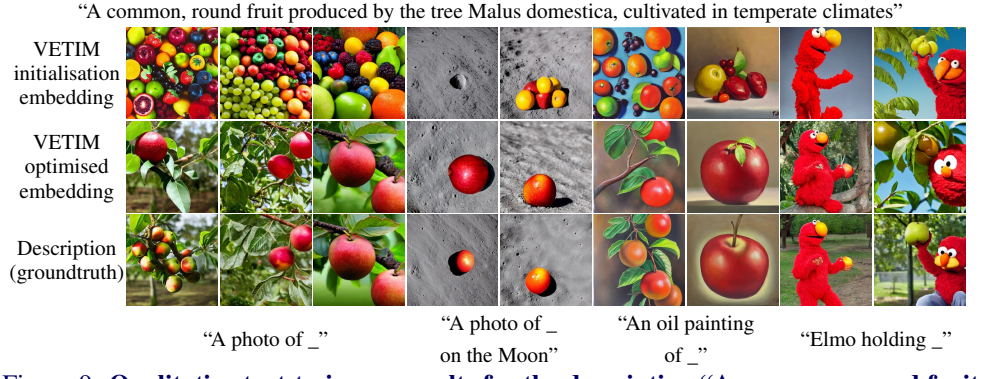

Figure 9: **Qualitative text-to-image results for the description "A common, round fruit produced by the tree Malus domestica, cultivated in temperate climates".** The rows *VETIM initialisation embedding* and *VETIM optimised embedding* contain images generated with the prompts "A photo of S_*", "A 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 token *fruits*. 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. The same seed was used to generate images in a given column.

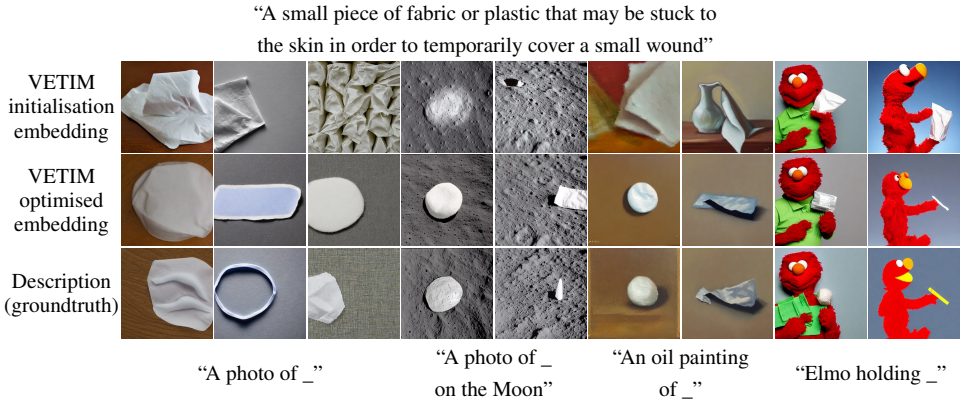

Figure 10: **Qualitative text-to-image results for the description "A small piece of fabric or plastic that may be stuck to the skin in order to temporarily cover a small wound".** Please refer to the caption of Figure 9 for details. For the row *VETIM initialisation embedding*, the closest embedding to the input description is, in this case, the one of the token *tissue*.

Figure 11: **Qualitative text-to-image results for the description "A collection of sheets of paper bound together to hinge at one edge, containing printed or written material, pictures, etc".** Please refer to the caption of Figure 9 for details. For the row *VETIM initialisation embedding*, the closest embedding to the input description is, in this case, the one of the token *paper*.

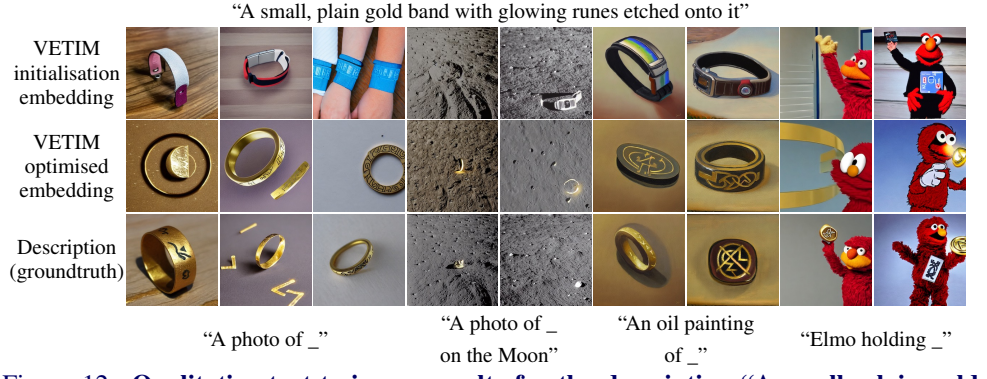

Figure 12: **Qualitative text-to-image results for the description "A small, plain gold band with glowing runes etched onto it".** Please refer to the caption of Figure 9 for details. For the row *VETIM initialisation embedding*, the closest embedding to the input description is, in this case, the one of the token *wristband*.

Figure 13: **Qualitative text-to-image results for the description "A frayed and patched wizard's hat with a mouth that moves and sings".** Please refer to the caption of Figure 9 for details. For the row *VETIM initialisation embedding*, the closest embedding to the input description is, in this case, the one of the token *snapback*.

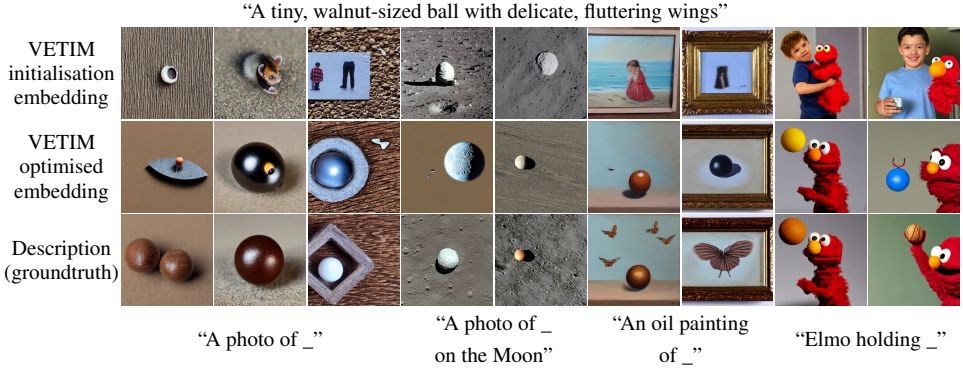

Figure 14: **Qualitative text-to-image results for the description "A tiny, walnut-sized ball with delicate, fluttering wings".** Please refer to the caption of Figure 9 for details. For the row *VETIM initialisation embedding*, the closest embedding to the input description is, in this case, the one of the token *smallest*.

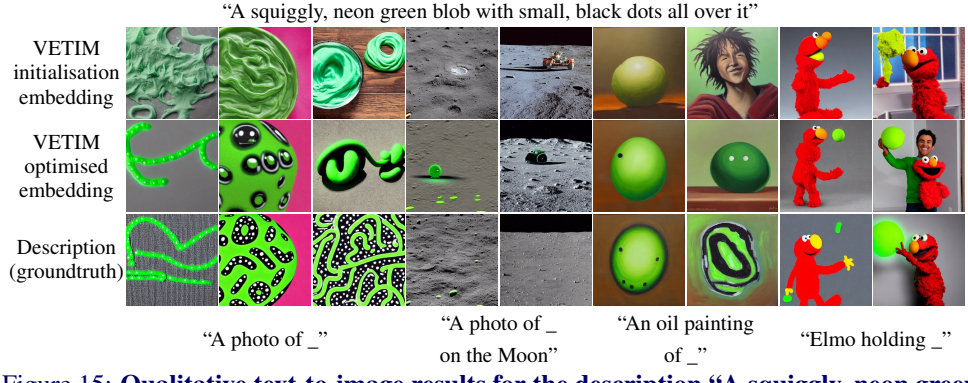

Figure 15: **Qualitative text-to-image results for the description "A squiggly, neon green blob with small, black dots all over it".** Please refer to the caption of Figure 9 for details. For the row *VETIM initialisation embedding*, the closest embedding to the input description is, in this case, the one of the token *slime*.

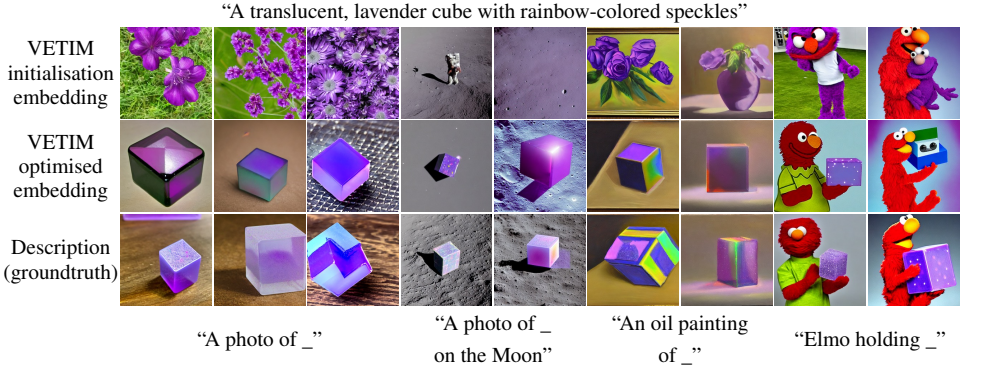

Figure 16: **Qualitative text-to-image results for the description "A translucent, lavender cube with rainbow-colored speckles".** Please refer to the caption of Figure 9 for details. For the row *VETIM initialisation embedding*, the closest embedding to the input description is, in this case, the one of the token *purple*.

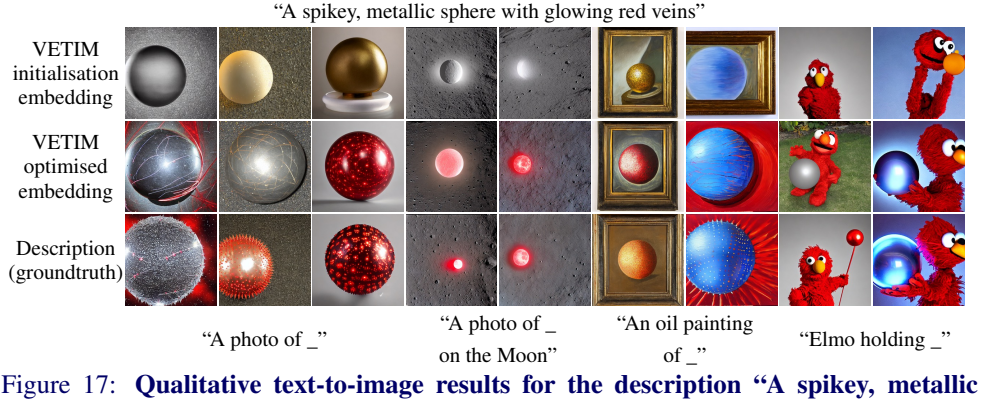

Figure 17: **Qualitative text-to-image results for the description "A spikey, metallic sphere with glowing red veins".** Please refer to the caption of Figure 9 for details. For the row *VETIM initialisation embedding*, the closest embedding to the input description is, in this case, the one of the token *orb*.

E Additional results and applications

E.1 Debiasing the vocabulary of a model

The results in Figure 6 of the main document were obtained with slightly different implementation details than described in Section A. We only used the format "`style_background`." to generate the list of templates $T[.]$ (see Paragraph **Complex templates** of Section A). In other words, the embedding of S_* was not trained within other objects. At the end of each optimisation, one per profession, the embedding of the token referring to the profession is over-written by the optimised embedding of S_*. Additionally, instead of initialising our optimisation with the closest token, the embedding v_* of S_* is initialised with the embedding of the original (biased) profession. Note we do not optimise the embedding of the biased term, but create and optimise a new embedding. We then erase the embedding of the biased term with the learned embedding.

We provide additional outputs and gender statistics in Figure 18. For *firefighter*, we generated 100 images with the prompt "A photo of a firefighter" with the original Stable Diffusion, 100 images with the prompt "A photo of a firefighter" with our customised Stable Diffusion where we overwrote the embedding of the token *firefighter*, and 100 images generated with the prompt "A photo of a firefighter if all genders can be a firefighter" with the original Stable Diffusion. We went manually over these generated images and annotated them as Male (M), Female (F) or Unknown/unsure (U). We did the same for *doctor*. The resulting gender statistics and the first 12 images are shown in Figure 18.

E.2 Improving interpretability with cross-attention maps

We provide additional results on DAAMs (Diffusion attentive attribution maps [5]), similar to Figure 7 of the manuscript. As can be seen in Figures 19, 20, and 21, images generated with tokens from VETIM are easier to interpret because fewer attention maps need to be analysed. Additionally, we compare the DAAM of the token S_* with the average of the DAAMs of the input description. The DAAM of the token S_* appears to better indicate which pixels are affected by the object.

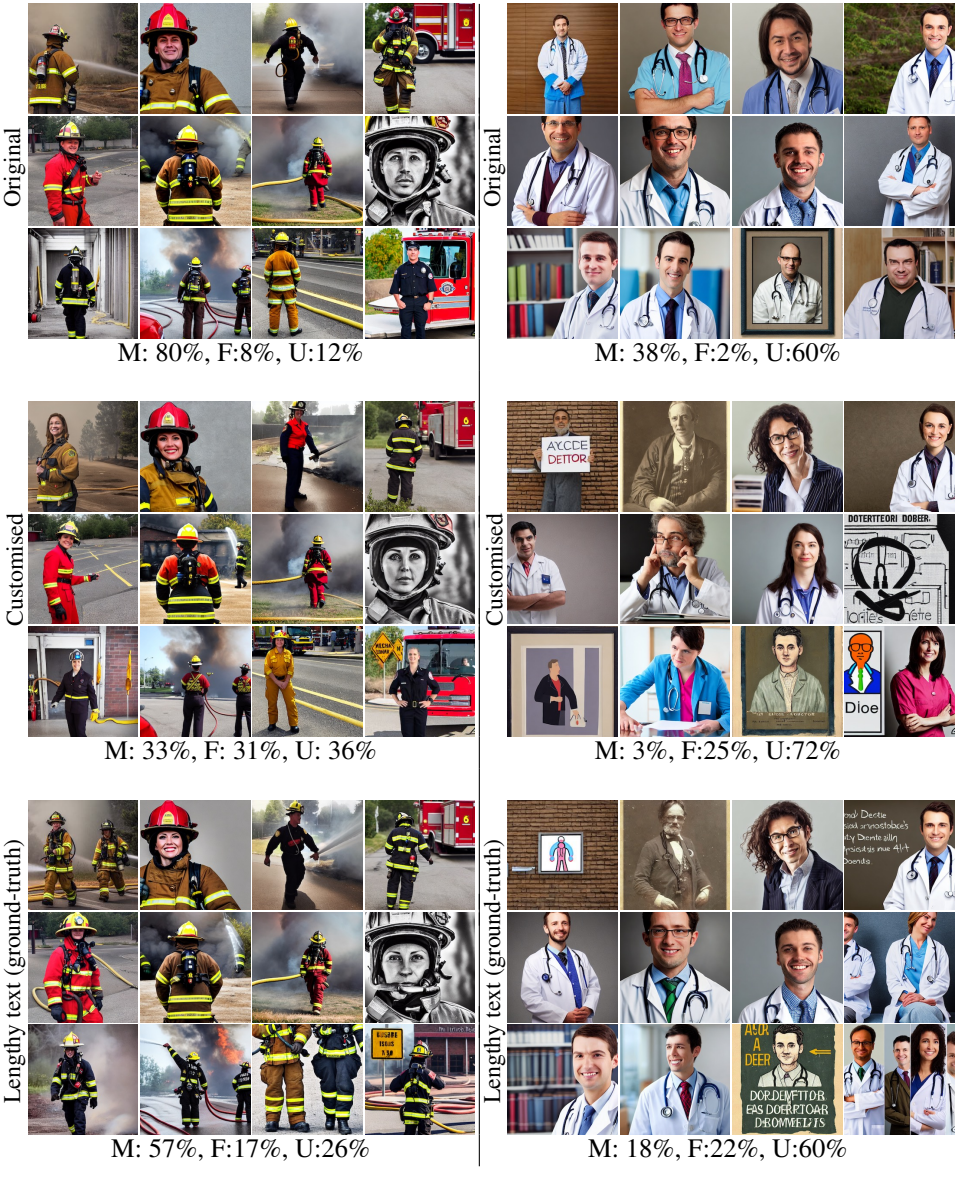

Figure 18: **Debiasing *firefighter* and *doctor*.** Left: The row *original* (resp. *customised*) contains 12 images generated with the prompt "A photo of a firefighter" with the original Stable Diffusion (resp. our customised Stable Diffusion where we overwrote the embedding of the token *firefighter*). The row *lengthy text* contains 12 images generated with the prompt "A photo of a firefighter if all genders can be a firefighter" with the original Stable Diffusion. For each row, genre statistics over 100 generated images are provided: M (Male), F (Female), U (Unknown or Unsure). Right: Same for "doctor".

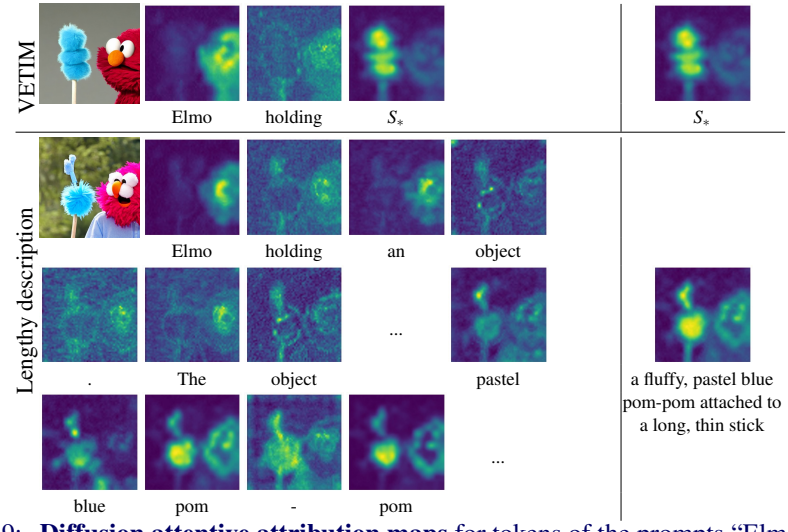

Figure 19: **Diffusion attentive attribution maps** for tokens of the prompts "Elmo holding S_*" and "Elmo holding an object. The object is a fluffy, pastel blue pom-pom attached to a long, thin stick.". We used VETIM to learn the embedding of S_* from the description "a fluffy, pastel blue pom-pom attached to a long, thin stick". On the right column, we give the DAAM of the token S_* from the prompt "Elmo holding S_*", and the average of the DAAMs of the tokens "a fluffy, pastel blue pom-pom attached to a long, thin stick" in the prompt "Elmo holding an object. The object is a fluffy, pastel blue pom-pom attached to a long, thin stick."

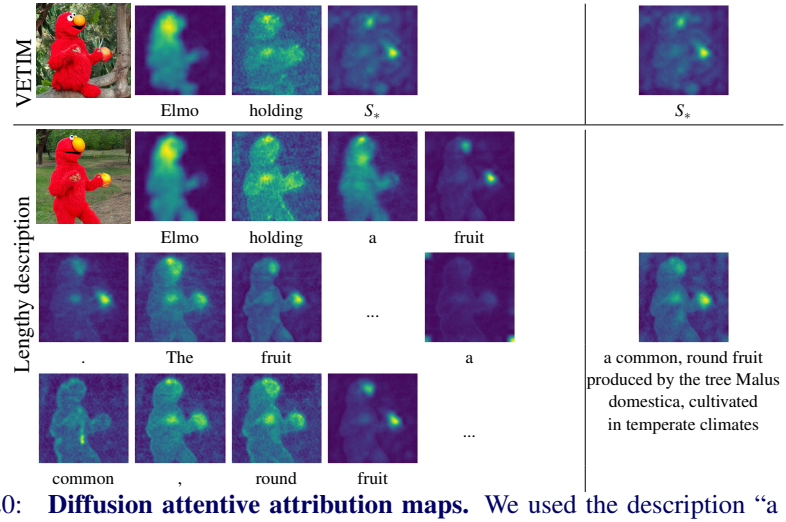

Figure 20: **Diffusion attentive attribution maps.** We used the description "a common, round fruit produced by the tree Malus domestica, cultivated in temperate climates" to learn the embedding of S_*. Please refer to the caption of Figure 19 for details.

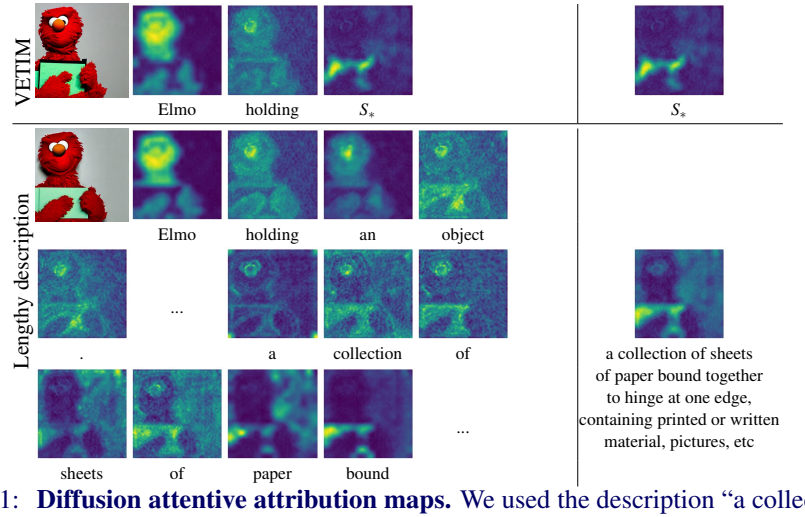

Figure 21: **Diffusion attentive attribution maps.** We used the description "a collection of sheets of paper bound together to hinge at one edge, containing printed or written material, pictures, etc" to learn the embedding of S_*. Please refer to the caption of Figure 19 for details.

E.3 Multiple concepts

Our method optimises concepts separately and iteratively, rather than multiple concepts simultaneously. It is possible to use prompts with complex compositions of concepts such as "X_* sitting on Y_* and holding Z_*" or to use word-swap with multiple concepts: "Man sitting on chair and holding cat" and Swap(*man*, *chair*, *cat* ; X_*, Y_*, Z_*). Using multiple concepts, similar to using long complex prompts with Stable Diffusion in general, tends to provide mixed results.

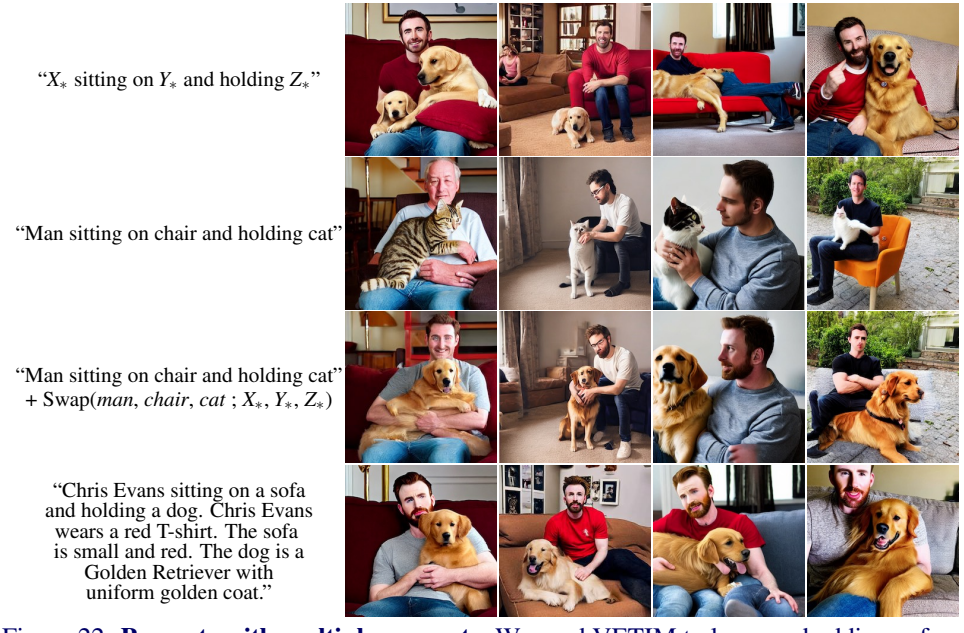

Figure 22: **Prompts with multiple concepts.** We used VETIM to learn embeddings of new tokens X_*, Y_* and Z_* from the descriptions "Chris Evans wearing a red T-shirt", "a small red sofa" and "a Golden Retriever with uniform golden coat". The first row shows 4 images generated from the prompt "X_* sitting on Y_* and holding Z_*". The second row shows 4 images generated from the prompt "Man sitting on chair and holding cat". The third row applies word-swap (`AttentionReplace`) to the images of the second row, swapping the tokens *man*, *chair*, and *cat* with X_*, Y_* and Z_*. The fourth row shows 4 images generated from the lengthy prompt indicated on the left.

F Use and limitations

Failure cases, importance of accurate text descriptions As can be seen in the different Figures, the images generated with the token optimised by VETIM do not always match very precisely the input description. For instance, in Figure 14, the object in the generated images does not contain wings as described in the input description "A tiny, walnut-sized ball with delicate, fluttering wings". We note that, in those failure cases, the same often happens when generating images from the full input text description. This highlights the importance of input descriptions in our approach. Our optimised embeddings are often as accurate as the

input descriptions when generating images, while maintaining the brevity of a single token.

Text-based versus Image-based vocabulary expansion Because we do not use sample images, VETIM do not learn to replicate the visual features of images, which can be done with methods such as Textual Inversion. VETIM can be used when we want to learn an embedding but cannot easily have images of the concept as input.

Future works may consider using VETIM as an extra supervision term in Textual-Inversion-like methods, potentially benefiting from the faster convergence of VETIM with the extra capability (learning new visual features not contained in the original model) of Textual Inversion.

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

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

[3] Gaoussou Youssouf Kebe, Padraig Higgins, Patrick Jenkins, Kasra Darvish, Rishabh Sachdeva, Ryan Barron, John Winder, Donald Engel, Edward Raff, Francis Ferraro, and Cynthia Matuszek. A Spoken Language Dataset of Descriptions for Speech-Based Grounded Language Learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

[4] OpenAI. ChatGPT, 2023.

[5] 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.