

The Hidden Language of Diffusion Models

Hila Chefer^{*1,2} Oran Lang¹ Mor Geva³ Volodymyr Polosukhin¹
 Assaf Shocher¹ Michal Irani^{1,4} Inbar Mosseri¹ Lior Wolf²
¹Google Research ²Tel-Aviv University ³Google DeepMind ⁴Weizmann Institute

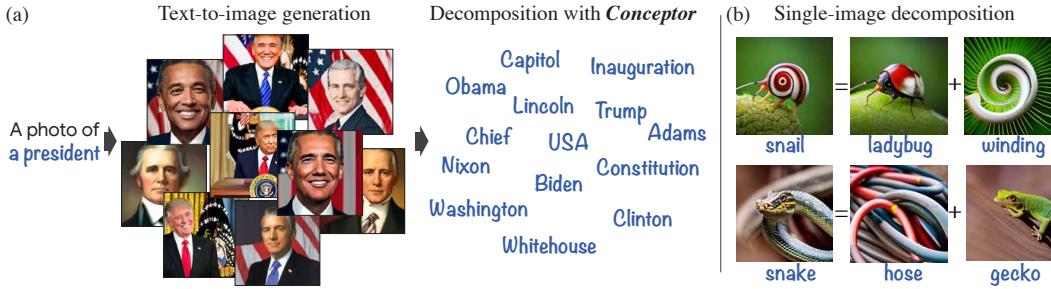


Figure 1: Concept decomposition using CONCEPTOR. (a) Given a concept of interest and a text-to-image model, we generate a set of images to visually represent the concept. CONCEPTOR then learns to decompose the concept into a small set of interpretable tokens, with the objective of reconstructing the generated images. The decomposition reveals interesting behaviors such as reliance on exemplars (e.g., “Obama”, “Biden”). (b) Our method enables various applications such as single-image decomposition to tokens and allows us to naturally visualize each token in the decomposition.

Abstract

Text-to-image diffusion models have demonstrated an unparalleled ability to generate high-quality, diverse images from a textual concept (e.g., “*a doctor*”, “*love*”). However, the internal process of mapping text to a rich visual representation remains an enigma. In this work, we tackle the challenge of understanding concept representations in text-to-image models by decomposing an input text prompt into a small set of interpretable elements. This is achieved by learning a pseudo-token that is a sparse weighted combination of tokens from the model’s vocabulary, with the objective of reconstructing the images generated for the given concept. Applied over the state-of-the-art Stable Diffusion model, this decomposition reveals non-trivial and surprising structures in the representations of concepts. For example, we find that some concepts such as “*a president*” or “*a composer*” are dominated by specific instances (e.g., “*Obama*”, “*Biden*”) and their interpolations. Other concepts, such as “*happiness*” combine associated terms that can be concrete (“*family*”, “*laughter*”) or abstract (“*friendship*”, “*emotion*”). In addition to peering into the inner workings of Stable Diffusion, our method also enables applications such as single-image decomposition to tokens, bias detection and mitigation, and semantic image manipulation. Our code will be available at: <https://hila-chefer.github.io/Conceptor/>.

^{*}The first author performed this work as an intern at Google Research.

1 Introduction

Consider a simple textual concept such as “*summer*” or “*love*”. What comes to mind? Naturally, we learn to associate concepts with combinations of other related concepts. For example, some may consider “*summer*” to be a composition of “*sun*”, “*beach*” and “*ice cream*”, and “*love*” to be the composition of “*hug*”, “*friendship*” and “*romance*”. Studies in the fields of cognitive science and natural language processing [9, 27] support the hypothesis that natural language concepts are represented by humans as a set of symbolic non-arbitrary links to other concepts. For example, as pointed out in [27], the human concept of “*cat*” is intuitively linked to other concepts such as “*ears*” and “*whiskers*”. However, while concept representations in the human brain and in natural language have been studied extensively [19, 20, 9], the same cannot be said about image generation models.

Recently, generative models have demonstrated unprecedented capabilities to create high-quality, diverse images based on textual descriptions [2, 10, 36, 39, 43]. However, as these models become increasingly expressive, our ability to understand *how* they map textual inputs into rich visual representations remains limited. In this work, we aim to demystify this internal process by interpreting the model’s latent representations of concepts using its textual space. Concretely, given a textual description of a concept, such as “*happiness*”, we propose to decompose the latent representation of the concept into a small set of interpretable tokens from the model’s vocabulary (see Fig. 1). To extract this set of features, our method, CONCEPTOR, learns a *pseudo-token*, which is a sparse linear combination of existing token embeddings, with the objective of reconstructing the concept images. Importantly, we show that this process results in a non-trivial, diverse set of learned tokens.

We use CONCEPTOR to analyze how state-of-the-art text-to-image diffusion models represent various concepts, including concrete (*e.g.*, “*a secretary*”) and abstract (*e.g.*, “*affection*”) concepts, and special cases of concepts with a double meaning (*e.g.*, “*a crane*”). Applying our method over the state-of-the-art Stable Diffusion model reveals interesting observations about the model’s behavior. First, as demonstrated in Fig. 1(b), our method can decompose every generated image to its own subset of underlying tokens. We find that similar to the hypothesis above, the generated images are often represented by direct combinations of related concepts that control different semantic aspects of the generated image. Second, we observe that some concepts such as “*a president*” or “*a composer*” are represented mostly by well-known instances from the concept (see Fig. 1 for example), such that the generated images are interpolations of those instances. We additionally find that, consistent with previous work [37], the model learns to mix the multiple meanings of homograph concepts, and leverages these meanings simultaneously when generating images from the concept. Finally, we demonstrate our method’s effectiveness in bias detection and semantic image manipulation.

To conclude, our work makes the following contributions:

- We develop a method to decompose a textual concept into a small set of interpretable tokens.
- We demonstrate single-image decompositions to determine what features caused the generation.
- We demonstrate interesting observations such as reliance on exemplars and entanglement of multiple meanings of a concept.
- We demonstrate fine-grained concept editing via manipulation of the coefficients in the decomposition. These manipulations assist in linking textual information to visual features.
- We demonstrate the detection of biases that are not easily observable visually. These observations raise important ethical questions regarding the social implications of leveraging these models.

2 Related Work

Early works studied text-guided image synthesis in the context of GANs [46, 55, 48, 52, 49]. More recently, impressive results were achieved with large-scale auto-regressive models [36, 50] and diffusion models [35, 31, 39, 43]. In the context of text-to-image diffusion models, a related line of work aims to introduce personalized concepts to a pre-trained text-to-image model by learning to map a set of images to a “token” in the text space of the model [11, 41, 22, 18]. However, these works do not investigate the inner representations of concepts but focus on concepts unfamiliar to the model.

A similar analysis to ours was conducted on concept representations in the context of language models [32, 23, 26, 28], often through projections to the vocabulary space [13, 12, 34]. Additionally, shred text-image representations such as CLIP [33] have been analyzed in the past [6, 5, 51] and have also been used to explain other models [30]. However, none of these works has been generalized to

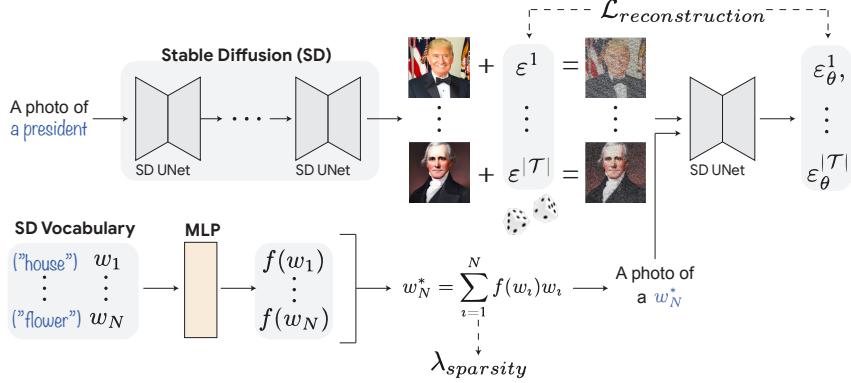


Figure 2: Illustration of the CONCEPTOR method. Given the concept of interest (e.g., “*a president*”), we generate 100 concept images. Next, a learned MLP network maps each word embedding w_i to a coefficient $f(w_i)$, and the pseudo token w_N^* is constructed as a linear combination of the vocabulary. We then add random noises $\varepsilon^1, \dots, \varepsilon^{|\mathcal{T}|}$ to the images, and use the model to predict the noise based on the text “*a photo of a <w_N^*>*”. We train the MLP with the objective of reconstructing the images ($\mathcal{L}_{reconstruction}$) and add a sparsity loss to encourage sparse coefficients ($\mathcal{L}_{sparsity}$).

image generation models. As far as we can ascertain, the closest effort to explaining text-to-image models is a simple visualization of the cross-attention maps per token in the prompt [15, 14, 4].

Finally, some works [37, 3] have attempted to investigate the images produced by text-to-image diffusion models, and have even found evidence of memorization [3]. However, these works rely entirely on the generated images and do not attempt to dissect the model’s inner representations. Unlike all the above, our method analyzes the inner workings of the model using its textual space, and our conclusions transcend those that can be obtained by simply examining the output images.

3 Method

3.1 Preliminaries: Latent Diffusion Models

We apply our method over the state-of-the-art Stable Diffusion (SD) model [39]. SD employs a denoising diffusion probabilistic model (DDPM) [44, 17] over an input latent vector $z_T \sim \mathcal{N}(0, 1)$ and gradually denoises it. Namely, at each timestep $t = T, \dots, 1$, the DDPM receives a noised latent vector z_t and produces a less noisy vector z_{t-1} , which serves as the input to the next step.

During the denoising process, the model is typically conditioned on a text encoding for an input prompt \mathcal{P} , produced by a frozen CLIP text encoder [33], which we denote by \mathcal{C} . The text encoder converts the textual prompt \mathcal{P} to a sequence of tokens, which can be words, sub-words, or punctuation marks. Then, the encoder’s vocabulary, $\mathcal{V} \in \mathbb{R}^{N,d}$, is used to map each token in the prompt to an embedding vector $w \in \mathbb{R}^d$, where d is the embedding dimension of the encoder, and N is the number of tokens in the vocabulary. The DDPM model is then trained to minimize the loss,

$$\mathcal{L}_{reconstruction} = \mathbb{E}_{z, \mathcal{P}, \varepsilon \sim \mathcal{N}(0, 1), t} [\|\varepsilon - \varepsilon_\theta(z_t, t, \mathcal{C}(\mathcal{P}))\|_2^2], \quad (1)$$

for,

$$z_t = \sqrt{\alpha_t} z + \sqrt{1 - \alpha_t} \varepsilon, \quad (2)$$

where ε_θ is a trained UNet [40], and $0 = \alpha_T < \alpha_{T-1} < \dots < \alpha_0 = 1$. In words, during training, the input image x is encoded to its corresponding latent vector z . A noise vector ε and a timestep t are drawn randomly. The noise vector ε is then added to the latent vector z as specified in Eq. 2, and the UNet is trained to predict the added noise ε .

3.2 CONCEPTOR

Our goal is to discover what features are used to encode a given concept c in a text-to-image diffusion model ε_θ . Formally, given a prompt \mathcal{P}^c for the concept c (e.g., “*a photo of a nurse*”), we learn a representation of the concept using the vocabulary \mathcal{V} . This representation is realized as a pseudo-token

$w^* \notin \mathcal{V}$ that is constructed as a weighted combination of a subset of tokens from \mathcal{V} , i.e.,

$$w^* = \sum_{i=1}^n \alpha_i w_i \quad \text{s.t.} \quad w_i \in \mathcal{V}, \quad (3)$$

where $n \leq N$ is a hyperparameter that determines the number of tokens to use in the combination.

Learning the set of n vocabulary elements w_i and their associated coefficients α_i is done separately for each concept c . To learn a meaningful pseudo-token w^* , we optimize it to reconstruct the images generated from \mathcal{P}^c , i.e., with the same objective as Eq. 1. We note that our method was purposefully constructed such that the optimization process mimics the training process of the model. This design is meant to encourage our pseudo-token to imitate the concept’s denoising process. In the following, we describe our method in detail, as illustrated in Fig. 2.

We begin by collecting a training set \mathcal{T} of 100 images generated from the concept. These images will be used for our reconstruction objective. Next, we compute the pseudo-token. Our method, CONCEPTOR, assigns a coefficient α for each word embedding w using a learned MLP on w . This way, the rich textual embedding space of CLIP is utilized in determining the coefficients. Specifically,

$$\forall w \in \mathcal{V} : \alpha = f(w) = W_2(\sigma(W_1(w))), \quad (4)$$

where σ is the ReLU non-linearity [1], and W_1, W_2 are linear mappings. Based on f , we compute $w_N^* = \sum_{i=1}^N f(w_i)w_i$. Note that this pseudo-token is not identical to the output token w^* since it contains all the tokens in \mathcal{V} . w^* is obtained by the top tokens from w_N^* , as described in Eq. 5.

Next, we turn to describe the reconstruction objective. To compute a reconstruction loss in the form of Eq. 1, we draw a random noise vector $\varepsilon \sim \mathcal{N}(0, 1)$, and a random timestep $t \in \{1, \dots, T\}$ for each of the images in the training batch, and noise the batch images according to Eq. 2. The reconstruction objective $\mathcal{L}_{\text{reconstruction}}$ is identical to the training objective specified in Eq. 1. However, x is now a noised version of a training image from \mathcal{T} , the weights of ε_θ are frozen, and the pseudo-token w_N^* , used by the prompt $\mathcal{P}^{w_N^*} = "a photo of a <w_N^*>"$, is learned. In other words, while the diffusion method trains the UNet for a frozen text prompt, we freeze the UNet and train the text prompt with the same objective. This is similar to personalization methods such as [11].

As mentioned, the pseudo-token w_N^* considers all the tokens in the vocabulary. However, for better interpretability, we wish to represent the input concept with a *small* set of $n << N$ tokens, where n is a hyperparameter that can be selected by the user. Notate by $w_1, \dots, w_n \in \mathcal{V}$ the tokens with the highest learned coefficients. We add a regularization loss to encourage the pseudo-token w_N^* to be dominated by these top n tokens, i.e.,

$$\mathcal{L}_{\text{sparsity}} = 1 - \text{cosine}(w^*, w_N^*). \quad (5)$$

This encourages the pseudo-token w^* , defined by the top n tokens in \mathcal{V} , to be similar to w_N^* , which is defined by the entire vocabulary. Our overall objective function is, therefore,

$$\mathcal{L} = \mathcal{L}_{\text{reconstruction}} + \lambda_{\text{sparsity}} \mathcal{L}_{\text{sparsity}}, \quad (6)$$

In our experiments, we set $\lambda_{\text{sparsity}} = 0.001$, $n = 50$. At inference time, we employ the per-concept MLP on the vocabulary \mathcal{V} to obtain the coefficients and consider the top $n = 50$ tokens to compose w^* , as specified in Eq. 3. Implementation details can be found in the supplementary materials.

3.3 Single-Image Decomposition

Given an image I that was generated by SD for a concept c , we wish to determine the subset of the tokens from the decomposition, w^* , that were involved in the generation of this specific image (see Fig. 1(b)). This is done via an iterative process over the tokens $w_j \in w^*$ as follows; at each step, we attempt to remove a single token from the decomposition, $w_j^* = \sum_{i \neq j} \alpha_i w_i$, and generate the corresponding image I_j with the prompt $\mathcal{P}^{w_j^*}$ and the same seed. Next, we use CLIP’s image encoder to determine if I_j is semantically identical to I . If the CLIP score of the two images is higher than 95, we remove w_j from w^* and continue to the next token. This criterion avoids incorporating tokens whose removal only causes minor non-semantic modifications to the image I (such as a slight shift in pose). This process is repeated for all tokens in w^* until no tokens are removed.

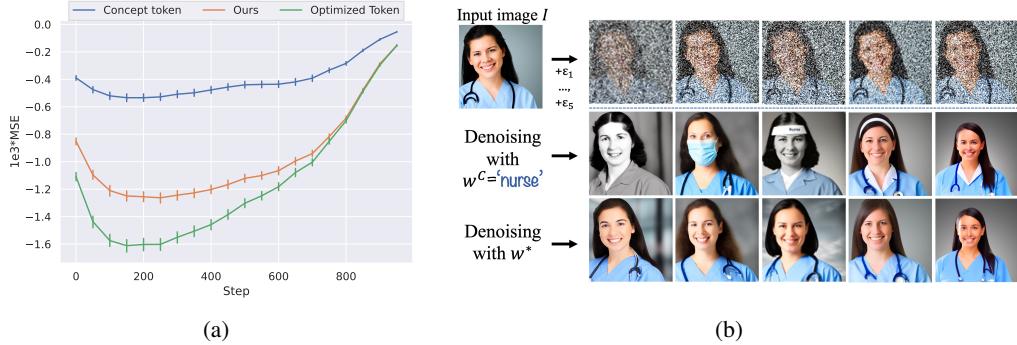


Figure 3: Denoising tests comparing the concept token, w^c , and our token, w^* . (a) Quantitative test on all 58 concepts using 100 test images per concept. For each timestep, we draw random noises for all images and compare the reconstruction with our pseudo token w^* , the concept token w^c , and w^o , a continuous token optimized for the same task (Optimized Token). We report the MSE after subtracting the reconstruction score of a random token, to reflect different levels of noise (lower is better). Note that the graph does not reflect a convergence of a reconstruction process, as timesteps are independent. Error bars are marked on each timestep (zoom in for better visibility). (b) Qualitative denoising examples. An image I is generated from the input concept “*a nurse*”, and different random noises are added $\varepsilon_1, \dots, \varepsilon_5$ (1st row). Denoising is done with w^c (2nd row) and with w^* (3rd row).

4 Experiments

We conduct experiments to show our method’s ability to produce meaningful decompositions for various concepts, from basic concepts (e.g., “dog”, “cat”) to rich concepts (e.g., “doctor”, “painter”) and abstract concepts (e.g., “happiness”, “fear”). Throughout this section, we note by w^c the token(s) corresponding to the concept c (e.g., for “*a nurse*”, w^c is *nurse*).

Data We construct a diverse dataset of 58 concepts, which are both concrete and abstract. For the concrete concepts, we consider the basic classes from CIFAR-10 [21], and the list of 28 professions from the Bias in Bios dataset [8]. For the abstract concepts, we use 10 basic emotions and 10 basic actions. A full list of all our considered concepts is provided in the supplementary materials.

Baselines As far as we can ascertain, our work is the first to tackle concept representation in text-to-image diffusion models. We, therefore, compare our method with reasonable and intuitive baselines. First, the most closely related method to ours is Hard Prompts Made Easy (PEZ) [47]. Given a set of input images, PEZ aims to learn a prompt such that when fed into SD, the resulting images will match the input images. This is done by prompt optimization to maximize the CLIP score between the text and the image. Second, we consider two baselines that leverage the state-of-the-art image captioning model BLIP-2 [24]: (ii) *BLIP-2 sentence*, extracts a single caption per concept. This is done by decoding the mean CLIP image embedding of the set of 100 training images \mathcal{T} generated for this concept. (iii) *BLIP-2 combination* creates one caption per each image $I \in \mathcal{T}$ and ranks the tokens obtained from all of the training set by their frequency across all such captions. Then, a single token is computed as the combination of the tokens weighted by their frequencies.

4.1 Motivation Experiments

In the following, we provide motivational experiments to better understand the capabilities of our method. We begin by addressing the seemingly unintuitive result that $w^* \neq w^c$. One may expect w^c , which generated the concept images, to be better than any other token in denoising them. However, this is not necessarily the case, since (1) w^* is optimized over linear combinations of tokens, including w^c . Therefore, given a successful optimization process, w^* is expected to perform at least as good as w^c , and potentially even better. (2) w^c generates each image using a specific initial random noise but is not guaranteed to be better in denoising the images after applying other random noises. Next, we compare the denoising quality of w^c and w^* quantitatively and qualitatively.

First, we wish to quantitatively verify that the denoising capabilities of w^* generalize to unseen images generated by the concept, beyond those used for training (Sec. 3). We begin by sampling a test set of

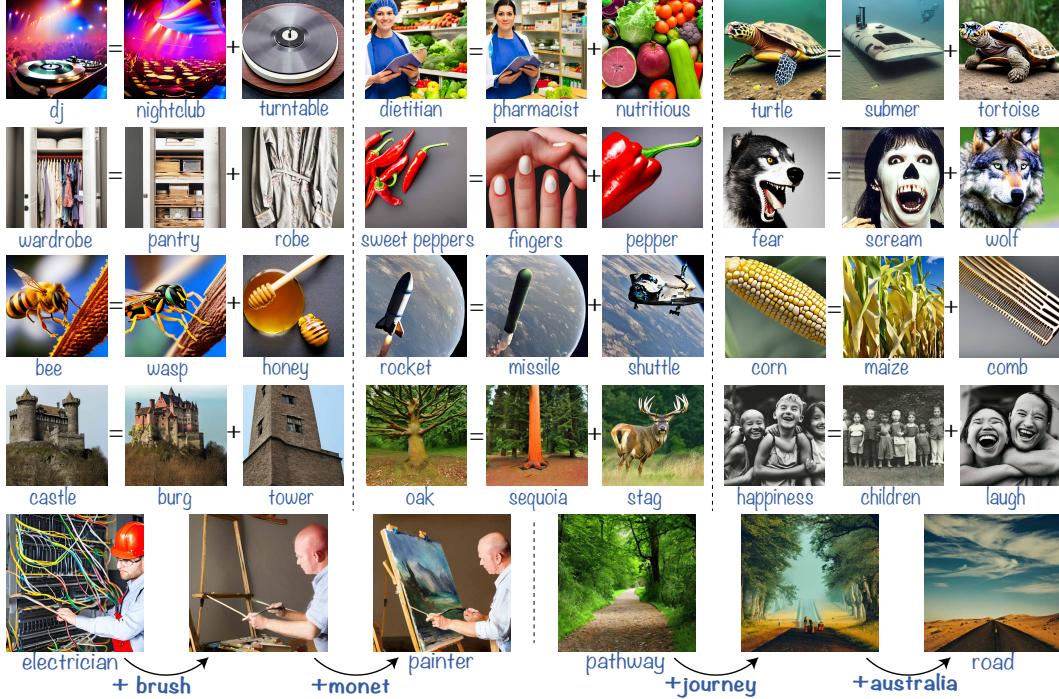


Figure 4: Single-image decompositions by CONCEPTOR. Each of the examples depicts an image generated by Stable Diffusion for the concept, and its corresponding decomposition. The top rows present images found to contain two concepts. The last row shows more complex mixtures by adding one token at a time from left to right (original image by SD on the right).

100 images for each concept, generated by w^c . Then, for each denoising step $t \in \{1, \dots, T\}$ and each test image, we draw a random noise and apply it as in Eq. 2. Finally, we test the reconstruction loss specified in Eq. 1 with the pseudo-token w^* compared to the concept token w^c . We additionally compare to w^o , a vector optimized with the same reconstruction objective on the entire continuous embedding space \mathbb{R}^d without restrictions, similar to [11]. Note that, unlike w^* , w^o does not offer interpretable information, but it provides a lower bound on the obtainable error. We observe that there is a large variance in the MSE score across timesteps. Latents in early steps are very noisy, and therefore obtain a very high loss (~ 0.8), while the last steps contain virtually no noise, and the MSE is very low ($\sim 10^{-3}$). Therefore, we compute a baseline score to normalize the scale by denoising the same images using a *random token* which serves as an upper bound for the MSE. The final MSE score for each token $w \in \{w^c, w^*, w^o\}$ is obtained by subtracting the MSE score of the random token from the MSE score of w , such that we maintain the convention that a lower score is better. Fig. 3(a) presents the results averaged across all 58 concepts, showing that the concept token w^c obtains a score worse than both w^* and the optimized token w^o , which obtains the best results. These differences are statistically significant, as shown by the error bars marked on every timestep. Evidently, by optimizing a token in a larger domain, we can outperform the original concept token in the denoising task.

Fig. 3(b) provides a qualitative comparison between w^c and w^* . An input image I generated by “*a photo of a nurse*” is noised and then denoised back from different denoising steps, using the concept token w^c and our pseudo-token w^* . As can be seen, there are cases where, given a different random seed, w^c does not preserve the features in the original image I (e.g., it adds hats, face masks, and black and white effects), while w^* does. Intuitively, this can be attributed to the rich representation learned by w^* , which can include both semantic and style features. Both experiments motivate the diversity of the learned decomposition. Since w^c is not necessarily optimal for Eq. 1, w^* learns additional features to improve the denoising quality. Thus, w^* balances two objectives—interpretability and better reconstruction.

Note that since $N >> d$, there are many linear combinations that yield w^* . However, due to the specific MLP-based structure and the sparsity constraints, the decomposition is stable, see supplementary materials for empirical evidence and additional experiments.

Table 1: Quantitative evaluation of CONCEPTOR and the baselines.

Method	CLIP [33] pairwise↑	LPIPS [53]↓	FID [16]↓	Token diversity [38]↑
PEZ [47]	74.5 ± 10.4	0.746 ± 0.04	152.95 ± 59.9	75.9 ± 1.2
BLIP-2 [24] sentence	76.2 ± 11.7	0.671 ± 0.1	155.78 ± 65.4	65.6 ± 1.6
BLIP-2 [24] combination	62.2 ± 18.4	0.873 ± 0.1	212.4 ± 97.5	51.4 ± 8.5
CONCEPTOR	86.7 ± 7.5	0.439 ± 0.1	106.89 ± 48.3	69.8 ± 3.4

Table 2: Qualitative comparison of the top tokens by our method and the leading baselines.

Concept	PEZ	BLIP-2 sentence	CONCEPTOR
Dog	viol, defam, qualifier, exoplan, beloved, dog	dog, black and white, face	Danes, pet, hybrid, Husky, pooch, Golden, Collie, Kennel
Composer	neh, barun, classical, pianist, nikol, themed	man, suit, tie, sitting, piano	Schneider, Millionaire, Beethoven, Preston, Schubert
Affection	inim, woolf, kaj, xoxo, tali, gossi, animals	dog, cat, hugging	Beautiful, Loving, Adorable, Puppies, Choose, Buddies

4.2 Qualitative and Quantitative Evaluation

Single-image decomposition Fig. 4 contains examples of decompositions obtained by our method over images generated by SD for various concepts, as described in Sec. 3.3. The first rows present examples of images that decompose into two tokens, while the last row contains more complex concepts. Observe that the first rows show non-trivial and profound semantic links between concepts. For example, “sweet peppers” are generated as peppers shaped like fingers, “rocket” is represented as a combination of missile and shuttle, and the generated image borrows its shape from the missile, and its appearance from the shuttle. Similarly, the “oak” borrows its shape from a sequoia tree and its appearance from the stag. Other concepts such as “happiness” combine related concepts such as children and laugh. These examples demonstrate that the model learns to link concepts beyond memorization, and can create associations based on semantic similarities such as shape and texture. In the last row, we present visual results of gradually adding features to construct the input image. Features are added from left to right and illustrate the semantic change induced by each token. As can be observed, each added token contributes different semantic features. For example, a “painter” is constructed as an electrician with a brush, and a Monet painting.

Comparison to baselines For each concept we test the reconstruction and diversity of the obtained decompositions. We use a test set of 100 seeds unseen during training to generate images with w^c and with each method. We employ three types of metrics: (i) *Pairwise Similarity*, which measures how well each method reconstructs the image generated by w^c . Different from Fig. 3, we no longer test denoising from a given noised input image, but rather the ability to generate concept images *from scratch*. Thus, w^c provides the ground truth data for this task. Reconstructing the concept images is crucial for two reasons. First, the decomposition must be faithful to the concept. Second, per-image reconstruction is necessary to enable single-image decomposition. We report the mean CLIP image-to-image similarity as well as the LPIPS [54] score. The former measures semantic similarity, while the latter measures image-level similarity. (ii) *Similarity between the distributions* is measured for each concept by computing the FID [16] score with respect to the images generated using w^c . Last, (iii) we employ SentenceBERT [38] to measure the average dissimilarity of each pair of tokens in the decomposition (*Word diversity*), to determine how rich the decomposition is.

The results, averaged across all 58 concepts, are reported in Tab. 1. As can be seen, in both pairwise metrics our method significantly outperforms the baselines by at least 10%. With respect to FID, CONCEPTOR also obtains the best results. We note that there is a large variance in the FID scores across concepts. This is since the model used to compute this score is InceptionNet [45] trained on ImageNet [42], which does not cover many of the concepts in our data (*e.g.*, humans). To provide a baseline for the best achievable FID, we compute the FID between the train and test images of each input concept and obtain a mean score of $101.05 (\pm 40.31)$, which is similar to the score obtained by our method. Considering the scores by SentenceBERT, CONCEPTOR is shown to provide diverse decompositions, where only PEZ supersedes our method. However, the prompts by PEZ contain a significant amount of uninterpretable tokens (see Tab. 2). Overall, CONCEPTOR achieves the best scores considering the two desirable but conflicting goals – interpretability and

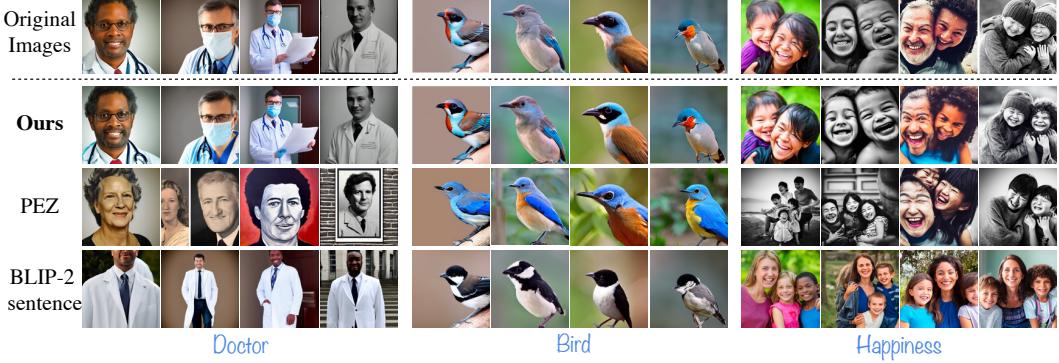


Figure 5: Reconstruction comparison to the baselines. For each concept (column) we generate the images from scratch starting from the same pure random noise with our method and all the baselines, and compare to the original concept images generated by Stable Diffusion (Original Images).

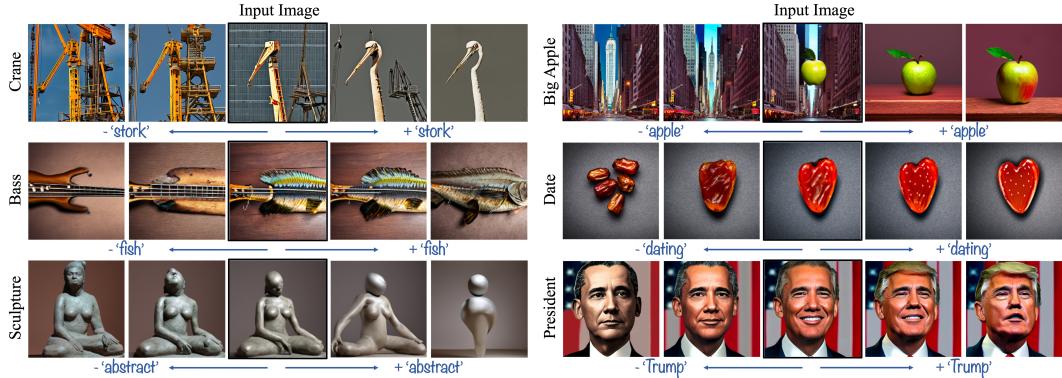


Figure 6: Feature visualizations. For each of the listed concepts, we manipulate a single textual token from the decomposition and observe its visual impact on the generated image.

reconstruction. Next, we conduct qualitative comparisons of CONCEPTOR to the baselines. Tab. 2 compares the textual decompositions, showing that CONCEPTOR learns a variety of features for each concept. Some concepts, such as “*a composer*”, “*a dog*” are dominated by instances of the concept (*e.g.*, Beethoven, Schubert), while others, such as “*affection*”, are represented by abstract and concrete tokens associated with the concept (*e.g.*, loving, puppies). In contrast, the baselines either produce decompositions that are not interpretable (PEZ) or oversimplistic (BLIP-2). Fig. 5 presents a comparison of the reconstruction by each method given the same 4 seeds. As can be observed, CONCEPTOR successfully preserves the image features, even when the concept entails fine features (*e.g.*, “*a bird*”). Conversely, the baseline methods either produce results that lack diversity (*e.g.*, BLIP-2 only generates black and white birds), or do not accurately embody all features of the concept (PEZ). The corresponding images and decompositions of the examples in Tab. 2 and Fig. 5 can be found in the supplementary materials, alongside experiments using exemplar-based concepts.

Feature Visualization Our method enables natural visualization for each token in the decomposition by manipulating its corresponding coefficient. By increasing the coefficient, the presence of the token becomes stronger, and vice versa. Fig. 6 presents examples of such visualizations.

First, following [37], we present examples of dual-meaning concepts (first, second row of Fig. 6). As can be seen, our decomposition allows us to visualize the impact of each of the meanings on the generated images. We observe that in some cases, such as “*a big apple*”, both meanings are generated in the original image, while other cases, such as “*crane*” generate a single object. Even in the latter cases, our method demonstrates that both meanings impact the generated image, implicitly. For example, when reducing the feature *stork* from “*crane*” we observe that the structure of the crane changes. Evidently, the dual meaning of the bird influenced the shape of the generated crane.

Next, we present an example of a feature that may appear unintuitive, *abstract* for the concept “*sculpture*”. We observe that this feature controls the level of detail in the generated image. Finally,

Table 3: Top tokens obtained by CONCEPTOR that reveal potential social insensitivity.

Concept	Decomposition
Secretary	clerk, prosecutor, teachers, wife, hostess, actress, womens, girl, ladies...
Opera singer	vanity, fat, obese, chiffon, soprano, overweight...
Pastor	Nigerian, directors, gospel, worship, tux...
Journalist	stranger, refugee, press, paparazzi, jews, tripod, photographing...
Drinking	cheating, millennials, liquid, blonde, pitcher, drunk, toast, smiling, booze...

Table 4: Ablation study of our method, conducted on the professions subset [8].

Method	CLIP [33] pairwise↑	LPIPS [53]↓	FID [16]↓	Token diversity [38]↑
CONCEPTOR	87.0 ± 5.5	0.45 ± 0.07	107.96 ± 31.0	69.7 ± 3.4
w/o MLP	78.0 ± 6.7	0.55 ± 0.06	142.88 ± 45.1	75.9 ± 3.0
w/o $\mathcal{L}_{sparsity}$	80.3 ± 11.6	0.52 ± 0.09	146.4 ± 63.4	73.2 ± 2.1
w/o $\mathcal{L}_{reconstruction}$	61.5 ± 8.6	0.65 ± 0.06	246.33 ± 73.8	68.3 ± 3.7
$n = 10$	82.9 ± 7.8	0.49 ± 0.11	129.41 ± 55.3	54.6 ± 9.4
$n = 100$	85.6 ± 6.9	0.47 ± 0.07	114.36 ± 39.7	72.8 ± 1.8
CLIP [33] top words	80.1 ± 9.9	0.513 ± 0.1	130.9 ± 57.2	66.3 ± 3.9

we present an example of an interpolation of notable instances. As can be observed, the token Trump controls the semantic similarity to Donald Trump and adds features that correspond to his identity.

Bias Detection and Mitigation One important capability of our method is bias discovery and mitigation. Text-to-image models, and specifically Stable Diffusion, have been shown to represent social biases [7, 29]. The decompositions obtained by CONCEPTOR can be used to discover such biases by analyzing the tokens in the decomposition. Tab. 3 lists some concepts that contain features that may be considered socially insensitive. Our method detects behaviors that are not necessarily observable visually such as millennials for “drinking”. These findings substantiate the need to conduct more research on concept representations in text-to-image models, as biases can impact the generation even if they are hard to detect visually. Using our method, users can also choose to generate debiased versions of these concepts by employing manipulations, as demonstrated in Fig. 6, which exemplifies our method’s ability to perform fine-grained concept editing. This manipulation enables the user to gradually decrease the biased tokens until an equal representation is achieved.

4.3 Ablation Study

We conduct an ablation study to examine the impact of each component on our method. First, we ablate the choice of employing an MLP to learn the coefficients and instead learn them directly. Next, we ablate each of our loss functions and the choice of $n = 50$. Last, we ablate our choice of vocabulary \mathcal{V} and instead extract the top 50 tokens by their CLIP similarity to the mean image.

The results are listed in Tab. 4. Replacing the MLP with a vector of weights is detrimental to all metrics except for token diversity. Both loss functions $\mathcal{L}_{sparsity}, \mathcal{L}_{reconstruction}$ are required to achieve good results. Without the reconstruction loss, the images are not linked to the decomposition, which severely damages both LPIPS and FID. Without the sparsity loss, the top 50 tokens do not necessarily reflect the learned token w_N^* and all metrics except for word diversity deteriorate. Additionally, observe that the performance decreases when employing $n = 10$, since the decomposition is not rich enough to represent all features. For $n = 100$, the results are similar to the full method, other than the diversity which improves a little. This indicates that CONCEPTOR is relatively stable to this parameter. Finally, when only considering the top words by CLIP similarity to the mean image, the performance decreases substantially, supporting the reliance of our method on a wide variety of tokens from the vocabulary, and not just the ones most correlated with the images.

5 Discussion and Limitations

While our method provides faithful concept decompositions, there are several limitations to consider. First, as mentioned in Sec. 4.2, the visual impact of the obtained tokens may not be completely aligned with their lexical meaning. For example, the token suffrage, which refers to a historical

movement for women’s rights, is highly influential when generating images of nurses. Visually, for the concept “*nurse*”, this token changes the style of the image to match that of a century ago.

Additionally, as demonstrated in Sec. 4.1, our pseudo-token w^* improves the denoising quality of w^c , which indicates the information added to w^* beyond the tokens of the concept. We find that for complex tokens with rich representations such as the professions or the abstract concepts, our learned token w^* improves the denoising quality significantly and learns a larger variety of features, while for simpler concepts such as those of CIFAR-10, the improvement over w^c is less significant. We refer the reader to the supplementary materials for further details.

6 Conclusions

How does a generative model perceive the world? Focusing on text-to-image models, we investigate the model’s internal knowledge of real-world concepts. We ask not *what* is generated but *why* these generations came to be. Our method, CONCEPTOR, proposes a decomposition scheme that mimics the model’s training process to uncover its inner representations. We find that, in correlation with humans, these models learn to link concepts to other related concepts. Via a per-image decomposition algorithm, we observe that the model leverages these connections in non-trivial ways that transcend the lexical meaning of the tokens. For example, in Fig. 4, “*sweet peppers*” are linked to “*fingers*” due to their structural similarity, in Fig. 1 “*snail*” is linked to “*winding*” due to the texture of its shell, *etc*. These findings demonstrate a generation process that is based on a semantic image-centered organization of the data, rather than on simple memorization. Furthermore, our method exposes less intuitive behaviors, such as the reliance on exemplars, mixing dual meanings of concepts, or non-trivial biases. In all cases, the novel paradigm allows us to peer into the inner workings of a model that, similarly to other foundation models, can still be considered an enigma.

7 Acknowledgements

This work was done during an internship at Google Research. We thank Shiran Zada, Ariel Ephrat, Omer Tov, and Roni Paiss for their early feedback and insightful discussions.

References

- [1] Abien Fred Agarap. Deep learning using rectified linear units (relu). *ArXiv*, abs/1803.08375, 2018.
- [2] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. *ArXiv*, abs/2211.01324, 2022.
- [3] Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. *ArXiv*, abs/2301.13188, 2023.
- [4] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models, 2023.
- [5] Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 397–406, October 2021.
- [6] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 782–791, June 2021.
- [7] Ching-Yao Chuang, Varun Jampani, Yuanzhen Li, Antonio Torralba, and Stefanie Jegelka. Debiasing vision-language models via biased prompts. *arXiv preprint arXiv:2302.00070*, 2023.
- [8] Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. Bias in bios: A case study of semantic representation bias in a high-stakes setting. 2019.

- [9] Jerry A. Fodor and Zenon W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28:3–71, 1988.
- [10] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. *arXiv preprint arXiv:2203.13131*, 2022.
- [11] 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. *arXiv preprint arXiv:2208.01618*, 2022.
- [12] Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. *arXiv preprint arXiv:2304.14767*, 2023.
- [13] Mor Geva, Avi Caciularu, Ke Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Conference on Empirical Methods in Natural Language Processing*, 2022.
- [14] Ligong Han, Yinxiao Li, Han Zhang, Peyman Milanfar, Dimitris Metaxas, and Feng Yang. Svdiff: Compact parameter space for diffusion fine-tuning. *arXiv preprint arXiv:2303.11305*, 2023.
- [15] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.
- [16] 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. In *NIPS*, 2017.
- [17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.
- [18] 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.
- [19] Markus Kiefer and Friedemann Pulvermüller. Conceptual representations in mind and brain: Theoretical developments, current evidence and future directions. *Cortex*, 48:805–825, 2012.
- [20] Markus Kiefer and Friedemann Pulvermüller. Connecting concepts in the brain by mapping cortical representations of semantic relations. *Nat Commun*, 2020.
- [21] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.
- [22] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. *arXiv preprint arXiv:2212.04488*, 2022.
- [23] Belinda Z. Li, Maxwell Nye, and Jacob Andreas. Implicit representations of meaning in neural language models. In *Annual Meeting of the Association for Computational Linguistics*, 2021.
- [24] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *ArXiv*, abs/2301.12597, 2023.
- [25] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022.
- [26] Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *The Eleventh International Conference on Learning Representations*, 2023.
- [27] Charles Lovering and Elizabeth-Jane Pavlick. Unit testing for concepts in neural networks. *Transactions of the Association for Computational Linguistics*, 10:1193–1208, 2022.
- [28] Charles Lovering and Elizabeth-Jane Pavlick. Unit testing for concepts in neural networks. *Transactions of the Association for Computational Linguistics*, 10:1193–1208, 2022.
- [29] Alexandra Sasha Luccioni, Christopher Akiki, Margaret Mitchell, and Yacine Jernite. Stable bias: Analyzing societal representations in diffusion models. *arXiv preprint arXiv:2303.11408*, 2023.

- [30] Jinqi Luo, Zhaoning Wang, Chen Henry Wu, Dong Huang, and Fernando De la Torre. Zero-shot model diagnosis. *arXiv preprint arXiv:2303.15441*, 2023.
- [31] 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.
- [32] Roma Patel and Ellie Pavlick. Mapping language models to grounded conceptual spaces. In *International Conference on Learning Representations*, 2022.
- [33] 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.
- [34] Ori Ram, Liat Bezalel, Adi Zicher, Yonatan Belinkov, Jonathan Berant, and Amir Globerson. What are you token about? dense retrieval as distributions over the vocabulary. *arXiv preprint arXiv:2212.10380*, 2022.
- [35] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *ArXiv*, 2022.
- [36] 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.
- [37] Royi Rassin, Shauli Ravfogel, and Yoav Goldberg. Dalle-2 is seeing double: Flaws in word-to-concept mapping in text2image models, 2022.
- [38] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing*, 2019.
- [39] 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.
- [40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [41] 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. *arXiv preprint arXiv:2208.12242*, 2022.
- [42] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2014.
- [43] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, Seyedeh Sara Mahdavi, Raphael Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *ArXiv*, abs/2205.11487, 2022.
- [44] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.
- [45] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2015.
- [46] Ming Tao, Hao Tang, Fei Wu, Xiao-Yuan Jing, Bing-Kun Bao, and Changsheng Xu. Df-gan: A simple and effective baseline for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16515–16525, 2022.
- [47] 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. *ArXiv*, abs/2302.03668, 2023.

- [48] 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.
- [49] Hui Ye, Xiulong Yang, Martin Takac, Rajshekhar Sunderraman, and Shihao Ji. Improving text-to-image synthesis using contrastive learning. *arXiv preprint arXiv:2107.02423*, 2021.
- [50] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022.
- [51] Tian Yun, Usha Bhalla, Ellie Pavlick, and Chen Sun. Do vision-language pretrained models learn composable primitive concepts?, 2023.
- [52] Han Zhang, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. Cross-modal contrastive learning for text-to-image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 833–842, 2021.
- [53] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.
- [54] 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.
- [55] 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.

Table 5: Decomposition consistency experiment. For each number of tokens ($k = 10, 25, 50$) we test the intersection between our learned top tokens and those learned by employing two *different* training sets of concept images, with different random initializations. The results demonstrate that the top tokens are consistent and robust across different training sets and seeds.

	No. of Tokens	Intersection
Concrete	Top 10	8.03 (80.3%) \pm 2.07
	Top 25	17.68 (70.7%) \pm 4.47
	Top 50	28.96 (57.9%) \pm 8.05
Abstract	Top 10	7.20 (70.2%) \pm 1.86
	Top 25	15.95 (63.8%) \pm 3.97
	Top 50	25.65 (51.3%) \pm 5.41

A Implementation Details

All of our experiments were conducted using a single A100 GPU with 40GB of memory. We train our per-concept MLP as specified in Sec. 3 of the main paper with 100 images generated from the concept using seed 1024 for a maximum of 500 training steps with a batch size of 6 (which is the largest batch size that could fit on our GPU). Additionally, we use a learning rate of $1e - 3$ (grid searched on 5 concepts between $1e - 2, 1e - 3, 1e - 4$). We conduct validation every 50 optimization steps on 20 images with a validation seed and select the iteration with the best CLIP pairwise similarity between the reconstruction and the concept images. We use the latest Stable Diffusion v2.1* text-to-image model employing the pre-trained text encoder from the OpenCLIP ViT-H model*, with a fixed guidance scale of 7.5.

Additionally, to filter out meaningless tokens such as emojis, punctuation marks, *etc.*, we consider the vocabulary to be the top 5,000 tokens by their CLIP similarity to the average of the training set. We note that this filtering method is fairly coarse, and meaningless tokens remain in the vocabulary, however, we find that this choice improves the convergence time of our MLP such that 500 iterations are enough to obtain meaningful decompositions. This choice is ablated in the main paper (see the CLIP top words ablation).

B Dataset

Our dataset can be split into two subsets: concrete concepts and abstract concepts. Each can be further split into two subsets. The concrete concepts are comprised of professions from Bias in Bios [8], and basic concepts from CIFAR-10 [21]. The abstract concepts are comprised of a list of 10 basic emotions and 10 basic actions.

The images for the concrete concepts were generated using the template “*a photo of a <concept>*”, the emotions with “*a photo of <concept>*”, and the actions with “*a person <concept>*”. The full concept lists for each subset are detailed below.

Bias in Bios concepts “professor”, “physician”, “attorney”, “photographer”, “journalist”, “nurse”, “psychologist”, “teacher”, “dentist”, “surgeon”, “architect”, “painter”, “model”, “poet”, “filmmaker”, “software engineer”, “accountant”, “composer”, “dietitian”, “comedian”, “chiropractor”, “pastor”, “paralegal”, “yoga teacher”, “dj”, “interior designer”, “personal trainer”, “rapper”

CIFAR-10 concepts “airplane”, “automobile”, “bird”, “cat”, “deer”, “dog”, “frog”, “horse”, “ship”, “truck”

Emotions “affection”, “anger”, “disgust”, “fear”, “happiness”, “honor”, “joy”, “justice”, “sadness”, “beauty”

*<https://github.com/Stability-AI/stablediffusion>

*https://github.com/mlfoundations/open_clip

Table 6: Qualitative comparison of the top tokens by our method and the leading baselines.

Concept	PEZ	BLIP-2 sentence	CONCEPTOR
Bird	valence, threats, aamir, mozambique, features	small, bird, black and white, head	teal, beak, peck, blue, brown, parrot, striped, dove
Doctor	dr, medicalportraits, charles, carly, article	man, white, coat, tie	physician, William, Peter, Robert, scrub, scientific
Happiness	ji, southkorea, laughing, happiness	woman, children, smiling, photo	children, laughing, families, dreams, angles, souls

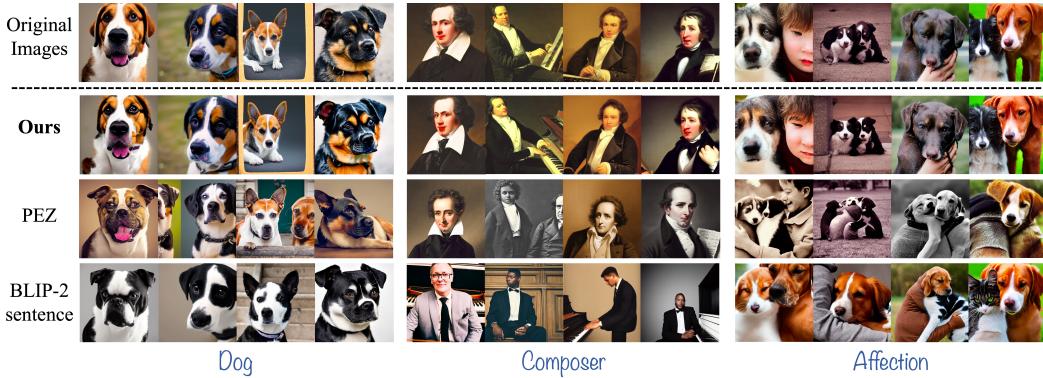


Figure 7: Reconstruction comparison to the baselines. For each concept (column) we generate the images from scratch starting from the same pure random noise with our method and all the baselines, and compare to the original concept images generated by Stable Diffusion (Original Images).

Actions “clapping”, “climbing”, “drinking”, “hugging”, “jumping”, “pouring”, “running”, “sitting”, “throwing”, “walking”

C Stability Experiments

We conduct experiments to empirically demonstrate our method’s stability, *i.e.*, verify that the same decomposition is obtained in runs that differ in the training set and initialization. For each concept, we generate 2 alternative training sets with different random seeds, in addition to our original training set, to verify the consistency of our results. For each alternative training set, we decompose the concept using our method as described in Sec. 3 of the main paper, with a different initialization for the MLP*. This process results in 3 decompositions of $n = 50$ tokens for each concept.

We then analyze the intersection of the top $k = 10, 25$, and 50 tokens between the original decomposition and each of the alternative decompositions. The concept intersection score for k is defined to be the average of the intersections with the two alternative sets. In other words, we calculate two intersection sizes for k : between the top k tokens of the original decomposition and the first alternative decomposition, and between the top k tokens of the original decomposition and the second alternative. The overall concept intersection score for k is the average of the two. Standard Deviation is computed across the concepts.

The average intersection scores across all concrete concepts and all abstract concepts are presented in Tab. 5. As can be seen, for the concrete concepts, an average of 8.03(80.3%) of the top 10 tokens are present in all the decompositions, even when considering an entirely different training set, indicating that the top tokens obtained by our method are stable. Additionally, when considering the top 25 tokens, an average of 17.68(70.7%) of the tokens are present in all decompositions, which is a large majority. We note that the bottom tokens are typically less influential on the decomposition, as they are assigned relatively low coefficients by the MLP. Accordingly, when considering all 50 tokens, an average of 28.96(57.9%) of the tokens appears in all decompositions. The results for the abstract concepts are slightly lower, yet demonstrate a similar behavior. Overall, the results demonstrate that

*Our code does not employ a fixed random seed, thus each run implies a different random initialization for the MLP and a different set of random noises and timesteps for the training images.

Table 7: Top tokens obtained by CONCEPTOR for concepts that rely on exemplars.

Concept	CONCEPTOR
Movie Star	Aubrey, Bourne, Lucille, Gloria, Marilyn, Monroe, Oswald
President	Obama, Trump, Biden, Nixon, Lincoln, Clinton, Washington
Rapper	Tupac, Drake, Khalifa, Weekend, Khaled, Eminem, Wayne

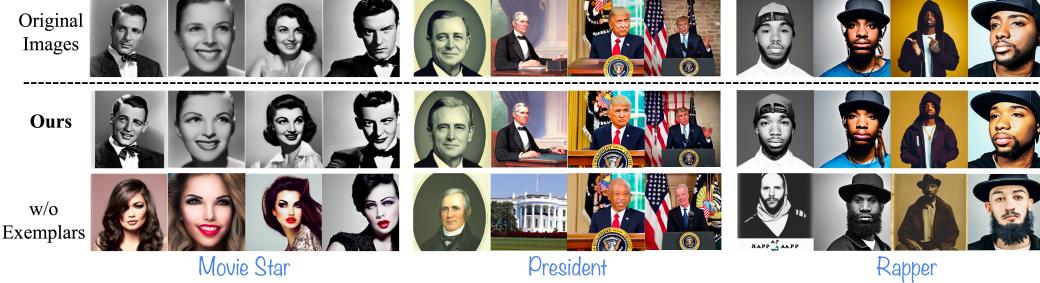


Figure 8: Examples of concepts that rely on famous instances. When removing the exemplars, the reconstruction quality is significantly harmed.

our method is relatively robust to different training sets and random seeds, such that even in the face of such changes, the top-ranked tokens remain in the decomposition.

D Additional Qualitative Comparisons

Tab. 6 and Fig. 7 present the complimentary qualitative results for Fig. 5 and Tab. 2 in the main paper, respectively. Tab. 6 shows a comparison between the textual decompositions by our method and the baselines for the concepts from Fig. 5 of the main paper. Fig. 7 compares the reconstruction quality of our method and the baselines for the concepts from Tab. 2 in the main paper.

As can be seen from Tab. 6, the same conclusions listed in the main paper hold. PEZ [47] produces results that are often uninterpretable, and BLIP-2 [25] produces over-simplistic captions that fail to capture the variety of features in the data. Fig. 7 demonstrates that our method is able to reconstruct the images in all cases, while the baselines fail to do the same.

E Representation by Exemplars

Tab. 7 and Fig. 8 present examples of concepts that rely on famous instances for their representations. For example, “*movie star*” is represented by names of famous actors such as Marilyn Monroe or Lucille Ball. Similarly, the concept “*president*” is dominated by American presidents such as Obama and Biden, and the concept “*rapper*” relies on famous rappers such as Tupac and Drake. To demonstrate the reliance on the exemplars in the decomposition, Fig. 8 shows the reconstruction by our method with and without the famous instance names from Tab. 7. As can be observed, the reconstruction quality heavily relies on the identities of the instances, and when we remove those instances the reconstruction is harmed significantly.

F Limitations- The Role of Context in Determining Semantics

As mentioned in the limitations section, a token in the decomposition could impact the resulting image in a manner that is not completely consistent with its lexical meaning. We find that the impact of each token significantly depends on the context provided by the other tokens in the decomposition.

Fig. 9 demonstrates such cases. For each concept (row) we visualize the effect of strengthening and weakening the token of interest on the input image (left). On the right side of the figure, we present the result of keeping only the token of interest and removing all others.

As can be seen, the influence of a token on the image does not necessarily correspond to its influence as a sole token. For example, the token *Washington* alone generates *Washington DC*, while in the

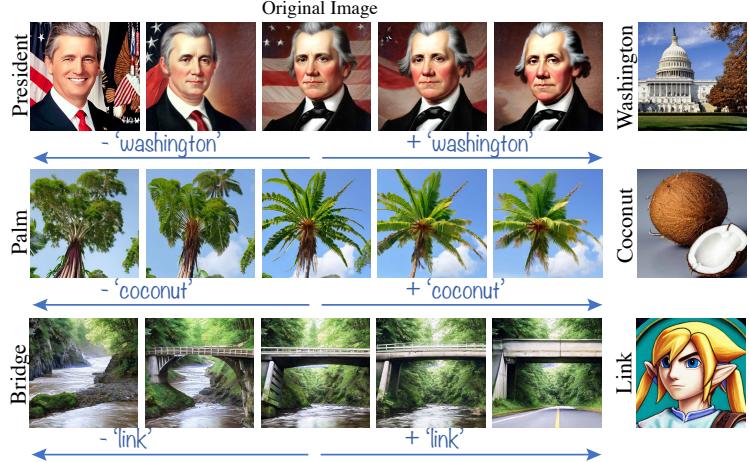


Figure 9: Tokens that impact the generated image differently from their lexical meaning. On its own, each token generates an image that is completely different than the input image (right). However, given the context of the decomposition, the effect of the token changes (left). *washington* generates Washington DC, while in the context of “*president*”, it adds the visual features of George Washington. *coconut* generates the fruit, while in the context of “*palm*”, it turns a tree into a palm tree. *Link* is a video game character, in the context of “*bridge*” it forms a bridge to *link* both parts of the image.

context of the decomposition for the concept “*president*”, this token adds/ removes the visual features that correspond to former American president George Washington. Similarly, the token *coconut* alone generates the fruit, while in the context of the concept “*palm*”, it turns the tree into a coconut tree (palm tree). Finally, *Link* is a video game character, therefore when generated on its own, it produces an image of the character. However, in the context of a bridge, this token adds a solid bridge between the two edges of the image.