

# $\mathcal{P}+$ : Extended Textual Conditioning in Text-to-Image Generation

Andrey Voynov<sup>1</sup>, Qinghao Chu<sup>1</sup>, Daniel Cohen-Or<sup>†1,2</sup>, Kfir Aberman<sup>1</sup>

<sup>1</sup>Google Research, <sup>2</sup>The Blavatnik School of Computer Science, Tel Aviv University

## Abstract

We introduce an Extended Textual Conditioning space in text-to-image models, referred to as  $\mathcal{P}+$ . This space consists of multiple textual conditions, derived from per-layer prompts, each corresponding to a layer of the denoising U-net of the diffusion model. We show that the extended space provides greater disentangling and control over image synthesis. We further introduce Extended Textual Inversion (XTI), where the images are inverted into  $\mathcal{P}+$ , and represented by per-layer tokens. We show that XTI is more expressive and precise, and converges faster than the original Textual Inversion (TI) space. The extended inversion method does not involve any noticeable trade-off between reconstruction and editability and induces more regular inversions. We conduct a series of extensive experiments to analyze and understand the properties of the new space, and to showcase the effectiveness of our method for personalizing text-to-image models. Furthermore, we utilize the unique properties of this space to achieve previously unattainable results in object-style mixing using text-to-image models.

## 1. Introduction

Neural generative models have advanced the field of image synthesis, allowing us to create incredibly expressive and diverse images. Yet, recent breakthroughs in text-to-image models based on large language-image models have taken this field to new heights and stunned us with their ability to generate images from textual descriptions, providing a powerful tool for creative expression, visualization, and design.

$\mathcal{P}+$  These text-to-image diffusion models use the encoded text as conditioning. We can refer to the conditioning space defined by the tokens embedding space of the language model as  $\mathcal{P}$  space. In other words,  $\mathcal{P}$  is the *textual-conditioning space*, where during synthesis, an instance  $p \in \mathcal{P}$  (after passing through a text encoder) is injected to all attention layers of a U-net, as illustrated in Figure 1 (left). In this paper, we introduce the *Extended*

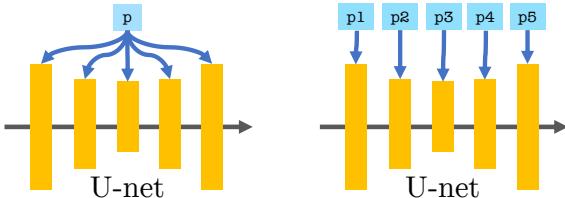


Figure 1.  $\mathcal{P}$  vs.  $\mathcal{P}+$ . Standard textual conditioning, where a single text embedding is injected to the network (left), vs. our proposed extended conditioning, where different embeddings are injected into different layers of the U-net (right).

Textual Conditioning space. This space, referred to as  $\mathcal{P}+$  space, consists of  $n$  textual conditions  $\{p_1, p_2, \dots, p_n\}$ , where each  $p_i$  is injected to the corresponding layer  $i$  in the U-net (see Figure 1 (right)).  $\mathcal{P}+$  space is more expressive, disentangled and thus provides better control on the synthesized image. As will be analyzed in this paper, different layers have varying degrees of control over the attributes of the synthesized image. In particular, the coarse layers primarily affect the structure of the image, while the fine layers predominantly influence its appearance.

The introduction of  $\mathcal{P}+$  textual conditioning space opens

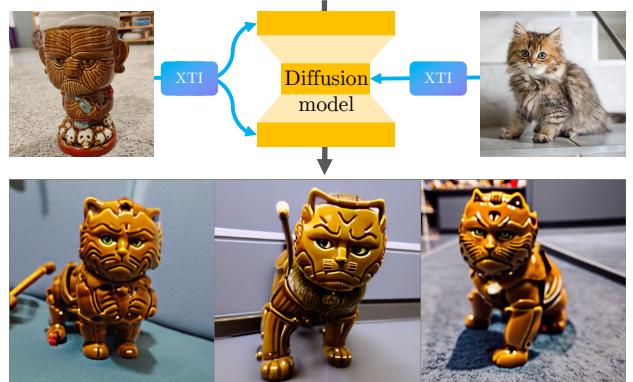


Figure 2. **Shape-Style Mixing in XTI.** The extended textual space allows mixing concepts learned from two separate extended textual inversions (XTIs). The inversion of the kitten (right) is injected to the coarse inner layers of the U-net, affecting the shape of the generated image, and the inversion of the cup (left) is injected to the outer layers, affecting the style and appearance.

<sup>†</sup>Performed this work while working at Google.

the door to a particularly exciting advancement in the domain personalization of text-to-image models [15, 45], where the model learns to represent a specific concept described in a few input images as a dedicated token. This learned token can then be employed in a text prompt to produce diverse and novel images related to the concept provided by the user. This technique of learning tokens is referred to as Textual Inversion (TI) [15].

In our work, we introduce Extended Textual Inversion (XTI), where we invert the input images into a set of token embeddings, one per layer, namely, inversion into  $\mathcal{P}^+$ . Our findings reveal that the expanded inversion process in  $\mathcal{P}^+$  is not only faster than TI, but also more expressive and precise, owing to the increased number of tokens that provide superior reconstruction capabilities. Remarkably, the improved reconstruction does not compromise editability, as demonstrated by our results.

Furthermore, we leverage the distinctive characteristics of  $\mathcal{P}^+$  to advance the state-of-the-art in object-appearance mixing through text-to-image generation. Specifically, we employ the insertion of inverted tokens of diverse subjects into the different layers to capitalize on the inherent shape-style disentanglement exhibited by these layers. This approach enables us to achieve previously unattainable results as shown in Figure 2.

We conduct extensive experiments and evaluation to demonstrate the effectiveness of the new space, analyzing its properties and showcasing its power for personalizing text-to-image models, object-style mixing, and more. Project page: <https://prompt-plus.github.io>

## 2. Related works

### 2.1. Extended Spaces in Generative Models

Exploring neural sub-spaces in generative models has been extensively explored, most notably in StyleGAN [21, 22]. The extended textual conditioning  $\mathcal{P}^+$  is reminiscent of StyleGAN’s extended latent space [1, 2], also commonly referred to as  $\mathcal{W}^+$ . Similar to  $\mathcal{W}^+$ ,  $\mathcal{P}^+$  is significantly more expressive, where instead of a single code shared by all layers, there is one per layer. However, while  $\mathcal{W}^+$  is an extended latent space, here the extended space relates to the textual conditions used by the network. It should be noted, though, that while  $\mathcal{W}^+$  is expressive, the extended code is less editable [50]. In contrast,  $\mathcal{P}^+$  remains practically as editable as  $\mathcal{P}$ . In addition, other sub-spaces lay within deeper and more disentangled layers [56] have been explored and exploited in various editing and synthesis applications [8].

In the case of text-to-image diffusion models, the denoising U-net, which is the core model of most of the text-to-image diffusion models, is usually conditioned by text prompts via a set of cross-attention layers [39, 42, 46]. In many neural architectures, different layers are responsible

for different abstraction levels [7, 22, 53, 59]. It is natural to anticipate that the diffusion denoising U-Net backbone operates in a similar manner, with different textual descriptions and attributes proving beneficial at different layers.

### 2.2. Text-Driven Editing

There has been a significant advancement recently in generating images based on textual inputs through Text-to-Image models [11, 39, 42, 46], where most of them exploit the powerful architecture of diffusion models [18, 42, 47, 48, 49].

In particular, recent works have attempted to adapt text-guided diffusion models to the fundamental problem of single-image editing, aiming to exploit their rich and diverse semantic knowledge of this generative prior.

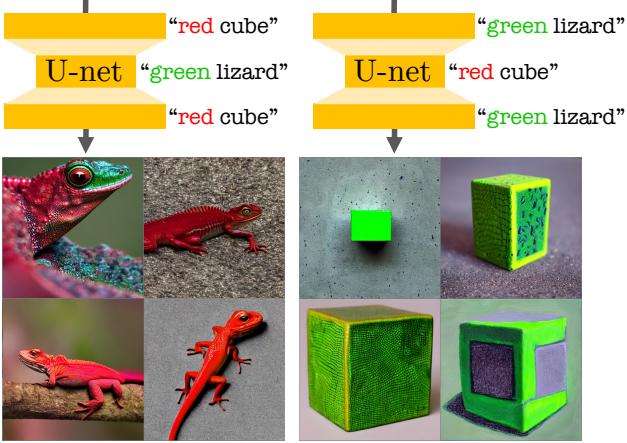
In a pioneering attempt, Meng et al. [31] add noise to the input image and then perform a denoising process from a predefined step. Yet, they struggle to accurately preserve the input image details, which were preserved by a user provided mask in other works [5, 4, 33]. DiffEdit [12] employs DDIM inversion for image editing, but to prevent any resulting distortion, it generates a mask automatically that allows background preservation.

Text-only editing approaches split into approach that supports to global editing [14, 25, 27, 37], and local editing [6, 54]. Prompt-to-prompt [17] introduces an intuitive editing technique that enables manipulation of local or global details by injecting internal cross-attention maps. To allow prompt-to-prompt to be applied to real images, Null-Text Inversion [32] is proposed as means to invert real images into the latent space of the diffusion model. Imagic [24] and UniTune [52] have demonstrated impressive text-driven editing capabilities, but require the costly fine-tuning of the model. The InstructPix2Pix [9], Plug-and-Play [51], and Parmar et al. [35] allow users to input an instruction or target prompt and manipulate real images accordingly, to achieve the desired edits.

### 2.3. Personalization

Synthesizing particular concepts or subjects which are not widespread in the training data is a challenging task. This requires an *inversion* process that given input images would enable regenerating the depicted object using a text-guided diffusion model. Inversion has been studied extensively for GANs [8, 13, 29, 57, 58, 60], ranging from latent-based optimization [1, 2] and encoders [40, 50] to feature space encoders [55] and fine-tuning of the model [3, 41, 34].

The notion of personalization of text-to-image models has been shown to be a powerful technique. Personalization of models [26, 45] in general or of text tokens [15] has quickly been adapted for various applications [23, 28]. In addition to their high computational cost, current methods face a clear-trade-off between learning tokens that ac-



**Figure 3. Per-layer Prompting.** We provide different text prompts (a precursor to  $\mathcal{P}+$ ) to different cross-attention layers in the denoising U-net. We see that color ("red", "green") is determined by the fine outer layers and content ("cube", "lizard") is determined by the coarse inner layers.

curately capture concepts vs. avoidance of overfitting. This can result in learned tokens that are overly tuned to the input images, thus limiting their ability to generalize to new contexts or generate novel variations of the concept.

Similar to TI, our approach does not require any fine-tuning or modification of the weights, thus, reduces the risk of overfitting and degrading the editability capabilites. In contrast, our inversion process into  $\mathcal{P}+$  is both faster and more precise, thanks to the greater number of tokens that improve reconstruction capabilities without sacrificing ed-ability.

### 3. Extended Conditioning Space

To engage the reader, we begin with a simple experiment on the publicly available Stable Diffusion model [43]. We partitioned the cross-attention layers of the denoising U-net into two subsets: coarse layers with low spatial resolution and fine layers with high spatial resolution. We then used two conditioning prompts: "red cube" and "green lizard", and injected one prompt into one subset of cross-attention layers, while injecting the second prompt into the other subset. The resulting generated images are provided in Figure 3. Notably, at the first run the model generates a red lizard, by taking the subject from the coarse layers' text conditioning, and appearance from the fine layers' conditioning. Similarly, in the second run it generates the green cube, once again taking the appearance from the fine layers and the subject from the coarse layers. This experiment suggests that the conditioning mechanism at different resolutions processes prompts differently, with different attributes exerting greater influence at different levels. With this in mind, our work aims to further explore this phenomena and its potential applications.

In the following parts, we introduce the Extended Textual Conditioning space ( $\mathcal{P}+$ ) and its key properties. We then detail how this space can be utilized to perform textual inversion for a given set of images.

#### 3.1. $\mathcal{P}+$ Definition

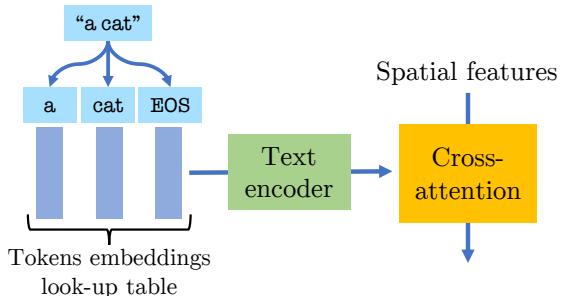
Let  $\mathcal{P}$  denote the *textual-conditioning space*.  $\mathcal{P}$  refers to the space of token embeddings that are passed into the text encoder in a text-to-image diffusion model. To clarify the definition of this space, we provide a brief overview of the process that a given text prompt undergoes in the model before being injected into the denoising network.

Initially, the text tokenizer splits an input sentence into tokens, with a special token marking the end of the sentence (EOS). Each token corresponds to a pre-trained embedding that is retrieved from the embedding lookup table. Subsequently, these embeddings are concatenated and passed through a pre-trained text encoder, then injected to the cross attention layers of the U-net model. In our work, we define  $\mathcal{P}$  as the set of individual token embeddings that are passed to the text encoder. The process of injecting a text prompt into the network for a particular cross-attention layer is illustrated in Figure 4.

We next present the *Extended Textual Conditioning* space, denoted by  $\mathcal{P}+$ , which is defined as follows:

$$\mathcal{P}+ := \{p_1, p_2, \dots, p_n\}, \quad (1)$$

where  $p_i \in \mathcal{P}$  represents an individual token embedding corresponding to the  $i$ -th cross-attention layer in the denoising U-net. Figure 1 illustrates the conceptual difference between the two spaces,  $\mathcal{P}$  (left) and  $\mathcal{P}+$  (right).



**Figure 4. Text-conditioning mechanism of a denoising diffusion model.** The prompt "a cat" is processed with a sentence tokenization, by a pretrained textual encoder, and fed into a cross-attention layer. Each of the three bars on the left represents a token embedding in  $\mathcal{P}$ .

With the definition of the new space, our diffusion model, previously conditioned on a single prompt  $U(x|t, p)$ , can now synthesize images in the extended space  $U(x|t, p_1, \dots, p_n)$ , where  $t$  denotes the denoising timestep.

As  $\mathcal{P}$  is a subspace of  $\mathcal{P}+$ , we naturally inquire about the advantages of synthesizing in the extended space. In

Section 4, we present an analysis of the properties of the new space, which showcases a higher degree of control over various attributes. Specifically, different layers are found to dominate different attributes, such as style, color, and structure.

A notable benefit of this space is its potential for enhancing textual inversion. We next demonstrate how the extended space can be utilized to represent subjects with greater fidelity, while maintaining the capability for editing.

### 3.2. Extended Textual Inversion (XTI)

Given a set of images  $\mathcal{I} = \{I_1, \dots, I_k\}$  of a specific subject, the goal of the Textual Inversion (TI) operation [15] is to find a representation of the object in the conditioning space  $\mathcal{P}$ . We next explain how we extend the TI approach and perform the inversion into  $\mathcal{P}+$ . This process is coined Extended Textual Inversion (XTI).

First, we add  $n$  new textual tokens  $\tau_1, \dots, \tau_n$  to the tokenizer model, associated with  $n$  new token embeddings lookup-table elements  $e_1, \dots, e_n$ . Then, similarly to [15], we optimize the token embeddings with the objective to predict the noise of a noisy images from  $\mathcal{I}$ , while the token embeddings are injected to the network.

Assuming that the denoising U-net is parameterized by a set of parameters denoted by  $\theta$ , and operates within the extended conditioning space as previously described, we define the reconstruction objective for the embeddings  $e_1, \dots, e_n$  that correspond to the tokens  $\tau_1, \dots, \tau_n$  as follows:

$$\mathcal{L}_{\text{XTI}} = \mathbb{E}_{\substack{P \sim \Pi, I \sim \mathcal{I}, \\ \varepsilon \sim \mathcal{N}(0,1), t}} \|\varepsilon - \varepsilon_\theta(I_t | t, P(\tau_1, \dots, \tau_n))\|_2^2$$

where  $I_t$  is the image  $I$  noised with the additive noise  $\varepsilon$  according to the noise level  $t$ . Once we operate with a latent diffusion model, we always suppose that  $I$  is a latent image representation. The new look-up table embeddings  $e_1, \dots, e_n$  that correspond to  $\tau_1, \dots, \tau_n$  are optimized w.r.t.  $\mathcal{L}_{\text{XTI}}$ . This optimization is applied independently to each cross-attention layer.

## 4. Experiments and Evaluation

In this section, we conduct an in-depth analysis of the various properties exhibited by the U-net cross-attention layers, and investigate how these characteristics are distributed across the layers. This analysis constitutes a motivation for the effectiveness of our  $\mathcal{P}+$  space. We then present a comprehensive evaluation of our proposed XTI approach for the personalization task, encompassing quantitative, qualitative, and user study analysis. For more details about the user study setting please refer to the supplementary material.

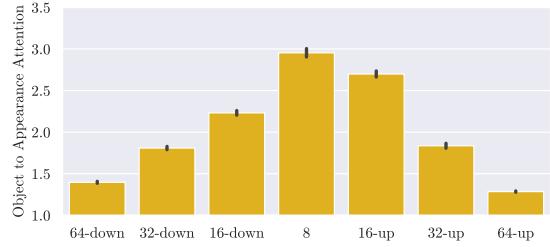


Figure 5. **Object-appearance attention ratio.** Mean ratio of attention features of the object token(s) and appearance token(s), per cross-attention layer.

In all of our experiments we use the Stable Diffusion 1.4 model [43]. It is built on top of CLIP [38], whose token embedding is represented by a vector with 768 entries, such that  $\mathcal{P} \subseteq \mathbb{R}^{768}$ . Stable Diffusion a latent diffusion model whose denoising U-net operates on an autoencoded image latent space. The U-net has four spatial resolution levels - 8x8, 16x16, 32x32, and 64x64. The 16, 32, and 64 resolution levels each have two cross-attention layers on the downward (contracting) path and three cross-attention layers on the upward (expansive) path. Resolution 8 has only 1 cross-attention layer. Thus there are a total of 16 cross-attention layers and 16 conditional token embeddings that comprise our  $\mathcal{P}+ \subseteq \mathbb{R}^{768 \times 16}$  space.

### 4.1. $\mathcal{P}+$ Analysis

**Cross-Attention Analysis** We first analyze how the distribution of the cross attention varies across layers. We create a list of 50 objects and 20 appearance adjectives (10 style descriptors and 10 texture descriptors, see supplementary material list). From these lists, we create 2000 ( $= 50 \times 20 \times 2$ ) prompts following the patterns "appearance object" and "object, appearance", and generate 8 images for each prompt using different seeds. We store the cross-attention values for each layer for only the object or appearance token(s), then average over the batch, spatial dimensions, and timesteps to get a ratio of attention on the object token(s) to attention on the appearance token(s). Figure 5 reports the corresponding ratios. The coarse layers (8, 16) attend proportionally more to the object token and fine layers (32, 64) attend more to the appearance token. This experiment gives us the intuition that coarse layers are more responsible for object shape and structure compared to the fine layers.

**Attributes Distribution** We further analyze how different cross-attention layers impact different image attributes (shape, color, etc.). To do so, we make use of the CLIP similarity metric [38] to quantify the contribution of each layer.

First, we divide the 16 cross-attention layers into 8 subsets, starting from the the empty set, followed by the middle



Figure 6. Visualization of mixed conditioning of the U-net cross-attention layers. The rows represent two different starting seeds and the columns represent eight growing subsets of layers, from coarse to fine. We start by conditioning all layers on "Blue car, impressionism" in the left column. As we move right, we gradually condition more layers on "Red house, graffiti", starting with the innermost coarse layers and then the outer fine layers. Note that *shape* changes ("house") take place once we condition the coarse layers, but *appearance* ("red") changes only take place after we condition the fine layers.

coarse layer and growing outwards to include the outer fine layers, and finally the full set (see Figure 7 for a visual explanation and the supplementary for the detailed list).

Next, we take three lists of `object`, `color` and `style` words and randomly generate prompts with the format "`color object, style`". For example, "`green bicycle, oil painting`" or "`red house, vector art`". We then randomly sample 64 pairs of these prompts. For every pair, we condition the aforementioned subset of layers on one prompt, and condition the complement set on the other prompt. We then generate 8 images with fixed seeds for each prompt-pair and subset.

Next, we measure the similarity of the output image to each `object`, `color` and `style` attribute with CLIP similarity. This measures the relative contribution of either conditioning prompt.

Figure 6 demonstrates this process for a single prompt pair and two image seeds. We start on the left column with all layers conditioned on the first prompt "`Blue car, impressionism`". As we move from left to right, we condition more layers from coarse to fine with the other prompt "`Red house, graffiti`". Note that even though we already condition some layers on "`Red house, graffiti`" in the middle column, the house only starts to appear red towards the end when the fine layers are also conditioned on the same prompt.

The results averaged over images and prompt pairs are shown in Figure 7. We see that at either extreme, the CLIP similarities are dominated by either prompt (represented as orange or blue). However, like the example in Figure 6, different prompt attributes demonstrate different behaviors in

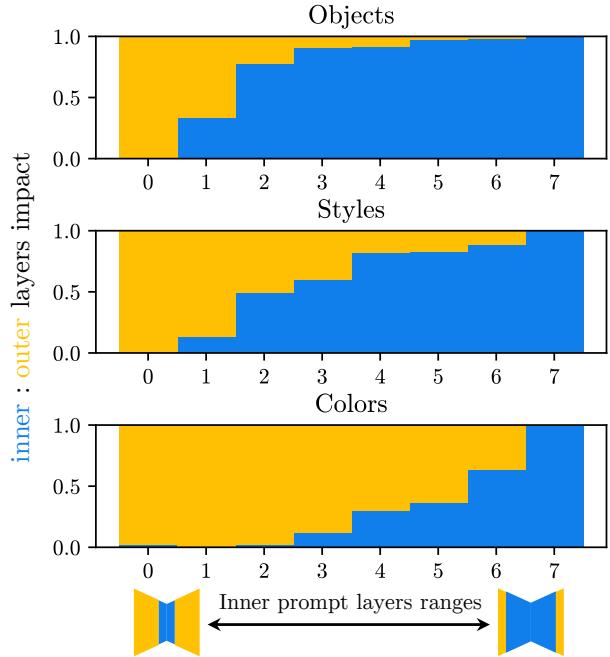


Figure 7. Relative CLIP similarities for `object`, `color` and `style` attributes, by subset of U-net layers. Orange represents the similarity to the first prompt, and blue represents similarity to the second. As we move from left to right, we gradually grow the subset of layers conditioned with the second prompt from coarse to fine.

between. We can see that it is sufficient to condition only the coarse layers for `object`, while `color` requires that we condition the full set of layers. `style` lies somewhere in-between. Thus, coarse layers determine the `object`

shape and structure of the image, and the fine layers determine the color appearance of the image. style is a more ambiguous descriptor that involves both shape and texture appearance, so every layer has some contribution towards it.

## 4.2. XTI Evaluation

Next, we evaluate our proposed XTI and compare our results to the original Textual Inversion (TI) [15]. We use a combined dataset of the TI dataset of 9 concepts, and the dataset from [26] with 6 concepts. For both datasets, each concept has 4-6 original images.

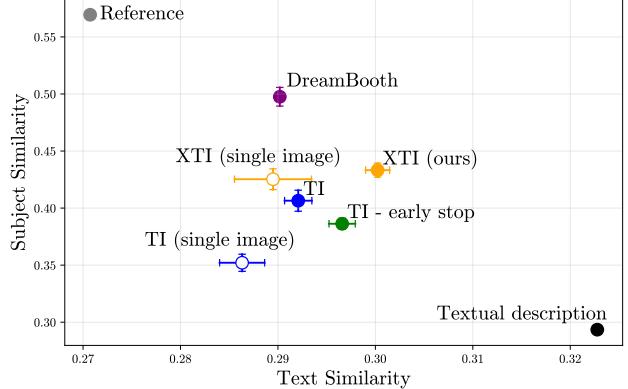
We focus on TI as a baseline because it is a model-preserving inversion approach that does not fine tune the model weights. These fine-tuning approaches like DreamBooth [45] and Custom Diffusion [26] explicitly embed the concept within the model’s output domain and thus have excellent reconstruction. However, they have several disadvantages. Firstly, they risk destroying the model’s existing prior (catastrophic forgetting). Secondly, they have several orders of magnitude more parameters. Recent work with Low-Rank Adaptation (LoRA) [20] reduces the number of fine-tuned parameters to a fraction, but this is still about  $\sim 100x$  more than XTI. Lastly, they are difficult to scale to multiple concepts since the fine-tuned parameters for each concept have to be merged.

### 4.2.1 Setup

We followed the batch size of 8 and performed 5000 optimization steps for Textual Inversion, consistent with the original paper. However, we opted to use a reduced learning rate of 0.005 without scaling for optimization, as opposed to the Latent Diffusion Model from [43] used in the original paper. In our experiments, Stable Diffusion with this learning rate worked better. For our proposed XTI, we used the same hyperparameters as for Textual Inversion, except for the number of optimization steps which we reduced to 500, **resulting in significantly faster convergence**. Both Textual Inversion and XTI shared all other hyperparameters, including the placeholder training prompts. On  $2 \times$ Nvidia A100 GPUs, the whole optimization takes  $\sim 15$  minutes for XTI compared to  $\sim 80$  minutes for TI.

### 4.2.2 Quantitative Evaluation

Following [15], to evaluate the editability quality of the inversions, we use the average cosine similarity between CLIP embeddings of the generated images and the prompts used to generate the images (*Text Similarity*). To measure the distortion of the generated images from the original concept (*Subject Similarity*), we use the average pairwise cosine similarity between ViT-S/16 DINO [10] embeddings of the generated images and the original dataset images.



**Figure 8. Comparison of Textual Similarity and Subject Similarity.** Textual Inversion (TI) [15], Extended Textual Inversion (XTI), DreamBooth [45]. We also evaluate the metrics for both multi-image and single-image inversion setups. For the latter, a subject is represented by a single image. The “Reference” label corresponds to images containing the subject images themselves, while the “Textual description” label used the given text description but replaced the explicit subject’s description (e.g. “a colorful teapot”). The standard error is visualized on the bars.

Compared to CLIP which is trained with supervised class labels, [45] argued that DINO embeddings better capture differences between images of the same class due to its self-supervised training. All the methods reported in Figure 8 are evaluated over 15 subjects from [15] and [26], each generated with 14 different prompts templates that place the concept in novel context (e.g. “A photograph of {} in the jungles”, see Section 8.2.3 in the supplementary for details). For each test concept and prompt we generated 32 images, making a total of  $15 \times 14 \times 32 = 6720$  images. We fix the generation seed across different methods.

In Figure 8 we report the evaluation of the proposed Extended Textual Inversion (XTI). Among Textual Inversion [15], as for comparison we also include DreamBooth [45] which is not a model-preserving method. Notably, XTI outperforms TI at both subject and text similarity despite using 10x fewer training steps. We also report TI using 500 optimization steps, which is the number of steps we use for XTI. This improves the Text Similarity because fewer number of optimization steps prevents the optimized token embedding from being out of distribution. However, it degrades reconstruction as measured by Subject Similarity.

We also report the inversion in a data-hungry setup, where the subject is represented with only a single image. Notably, even in this extreme setting the proposed XTI performs better than multi-image TI in terms of subject similarity (see Section 4.3 for details).



Figure 9. **Textual Inversion (TI) vs. Extended Textual Inversion (XTI).** Column 1: Original concepts. Column 2: TI results. Column 3: XTI results. It can be seen that XTI exhibits superior subject and prompt fidelity, as corroborated by the results of our user study.

#### 4.2.3 Human Evaluation

Figure 9 shows a visual comparison of our XTI approach with the original TI. Our method demonstrates less distortion to the original concept *and* to the target prompt.

To assess the efficacy of our proposed method from a human perspective, we conducted a user study. The study, summarized in Table 1, asked participants to evaluate both Textual Inversion (TI) and Extended Textual Inversion (XTI) based on their fidelity to the original subject and the given prompt. The results show a clear preference for XTI for both subject and text fidelity.

Method	Subject Fidelity	Text Fidelity
Textual Inversion	24%	27%
XTI (Ours)	<b>76%</b>	<b>73%</b>

Table 1. User study preferences for subject and text fidelity for TI and XTI. See supplementary material for more details.

#### 4.3. Single Image Inversion

The Extended Textual Inversion also appears to be very effective in a data-hungry setup, when a target subject is represented with a single image. As for single image training for all the runs we reduce learning rate to 0.001 to better prevent overfitting. Figure 10 provides visual comparison of TI and XTI inversions in this single image setting. We omit single-image DreamBooth results from Figure 10 and 8 due to its comparatively poor performance, namely Text Similarity of 0.25 and Subject Similarity of 0.40. In particular, we found DreamBooth in this single-image setting to be prone to overfitting and difficult to optimize.

#### 4.4. Embedding Density

As the textual embeddings inverted with XTI have better editability properties compared to the original TI, this suggests that these tokens are better aligned with the original tokenizer look-up table embedding, which represents the

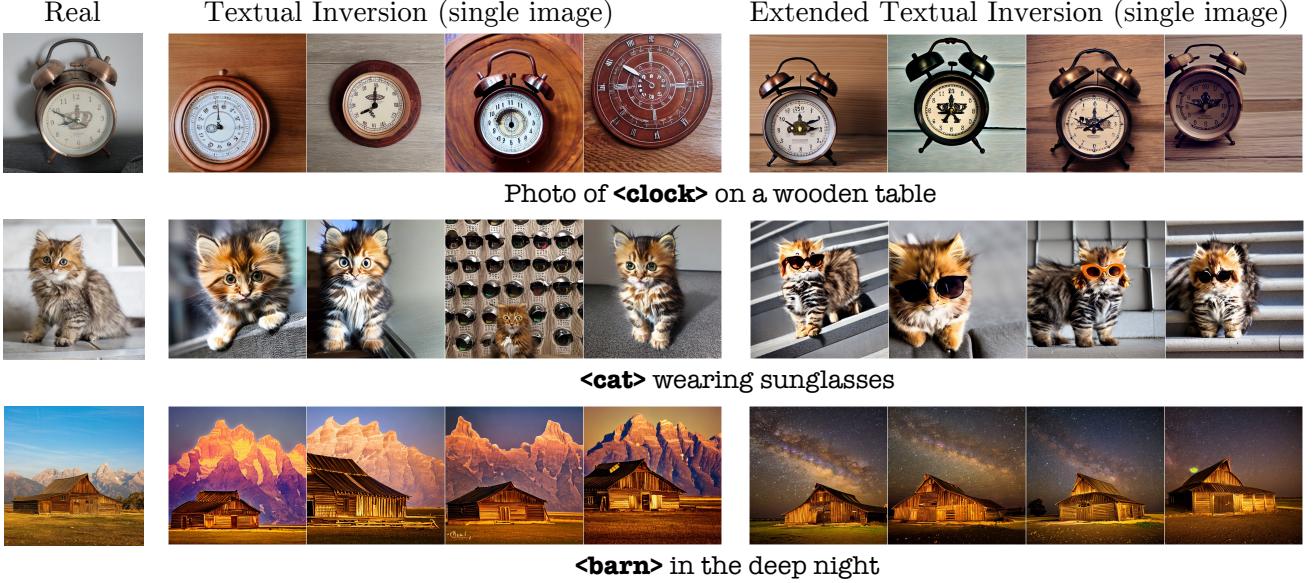


Figure 10. **Single Image Textual Inversion (TI) vs Single Image Extended Textual Inversion (XTI).** *Column 1:* Original concepts. *Column 2:* TI results. *Column 3:* XTI results. It can be seen that XTI exhibits superior subject and prompt fidelity and produce pleasurable results even when trained on a single image.

manifold of natural language embedding. To quantify this intuition, we evaluate the density of the newly-optimized tokens with respect to the original “natural” tokens look-up table embeddings. We perform kernel-based density estimation (KDE) in the look-up table tokens embeddings space independently for each dimension. Let us define  $\mathcal{E}$  to be the set of all original tokens look-up table embeddings, before adding the extra optimized token(s). Assuming that  $\mathcal{E}$  is sampled from some continuous distribution, one can define the approximation of its density function at a point  $x$  as:

$$\log p_{\mathcal{E}}(x) \approx \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} K(x - e), \quad (2)$$

where  $K$  is the Gaussian kernel density function [36, 44]. As for the embeddings optimized with original TI, this quantity always appears to be significantly smaller compared to the densities at the original embeddings  $\mathcal{E}$ . Figure 11 illustrates the original tokens density distribution, and the textual inversion tokens densities. This demonstrates that the proposed approach provides embeddings that are closer to the original distribution, enabling a more natural reconstruction and better editability.

## 5. Style Mixing Application

As we showed earlier, different layers of the denoising U-net are responsible for different aspects of a synthesized image. This allows us to combine the *shape* of one inverted concept with the *appearance* of another inverted concept. We call this Style Mixing.

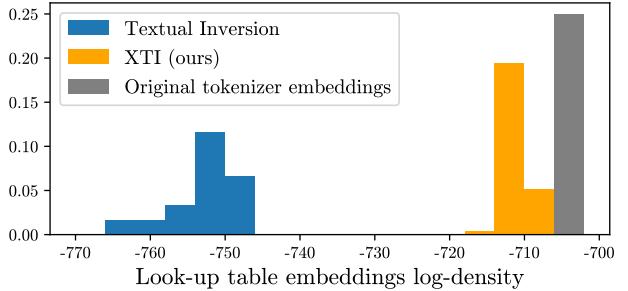


Figure 11. Estimated log-density of the original look-up table token embeddings (gray), embeddings optimized with textual inversion (blue), and embeddings optimized with XTI (orange). Our method demonstrates a more regular representation which is closer to the manifold of natural words.

Let us consider two independent XTI inversions of two different concepts. We can combine the inversions by passing tokens from different subject to different layers, as illustrated in Figure 2. This mixed conditioning produces an image with a coarse geometry from the first concept, and appearance from the second concept. Formally, we are given two extended prompts:  $\{p_1, \dots, p_n\}$ , and  $\{q_1, \dots, q_n\}$ . We form a new extended prompt  $\{p_1, \dots, p_k, q_{k+1}, \dots, q_K, p_{K+1}, \dots, q_n\}$  with the separators  $1 \leq k < K \leq n$ .

Our observations indicate that the optimization of XTI with an additional density regularization loss term indicated in 2 enhances its ability to mix objects and styles, without compromising the quality of the inversion output. More de-

tails are provided as supplementary material.

Figure 12 demonstrates combining the "skull mug" and "cat statue" concepts from [15]. Different rows of the plot correspond to different blending ranges  $k, K$ . From top to bottom, we gradually expand it from the middle coarse layer to all the cross-attention layers. This range ( $k, K$ ) gives the control over the amount of details we want to bring from one inversion to another. By varying  $k$  and  $K$ , we can adjust the contributions of the second subject appearance to the first.

Figure 13 shows a variety of examples generated with this method. Both shape and appearance are inherited remarkably well. More illustrations are provided in supplementary.

Figure 14 provides a qualitative comparison between our XTI-based style mixing and baselines of TI [15] and DreamBooth [45]. The results demonstrate that our approach outperforms the baselines significantly, both in terms of preserving the sources' fidelity and disentangling the attributes.



Figure 12. **Style Mixing in  $\mathcal{P}+$ .** Rows are generated by varying the degree of mixing by adjusting the proportion of layers conditioned on either of the two  $\mathcal{P}+$  inversions.

## 6. Conclusions, Limitations, and Future work

We have presented,  $\mathcal{P}+$ , an extended conditional space, which provides increased expressivity and control. We have analyzed this space and showed that the denoising U-net

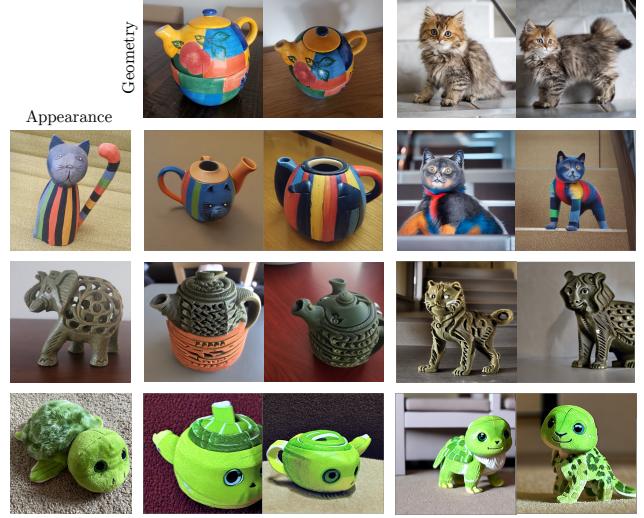


Figure 13. **More Style Mixing examples.** Top row: Shape source concepts. Left column: Appearance source concepts.

demonstrates disentanglement, where different layers exhibit different sensitivity to shape or appearance attributes

The competence of  $\mathcal{P}+$  is demonstrated in the Textual Inversion problem. Our Extended Textual Inversion (XTI) is shown to be more accurate, more expressive, more controllable, and significantly faster. Yet surprisingly, we have not observed any reduction in editability.

The performance of XTI, although impressive, is not flawless. Firstly, it does not perfectly reconstruct the concept in the image, and in that respect, it is still inferior to the reconstruction that can be achieved by fine-tuning the model. Secondly, although XTI is significantly faster than TI, it is a rather slow process. Lastly, the disentanglement among the layers of U-net is not perfect, limiting the degree of control that can be achieved through prompt mixing.

An interesting research avenue is to develop encoders to invert one or a few images into  $\mathcal{P}+$ , possibly in the spirit of [16], or to study the impact of applying fine-tuning in conjunction of operating in  $\mathcal{P}+$ .

## 7. Acknowledgement

We thank Eric Tabellion, Rinon Gal, Miki Rubinstein, Matan Cohen and Jason Baldridge for their valuable inputs that helped improve this work.

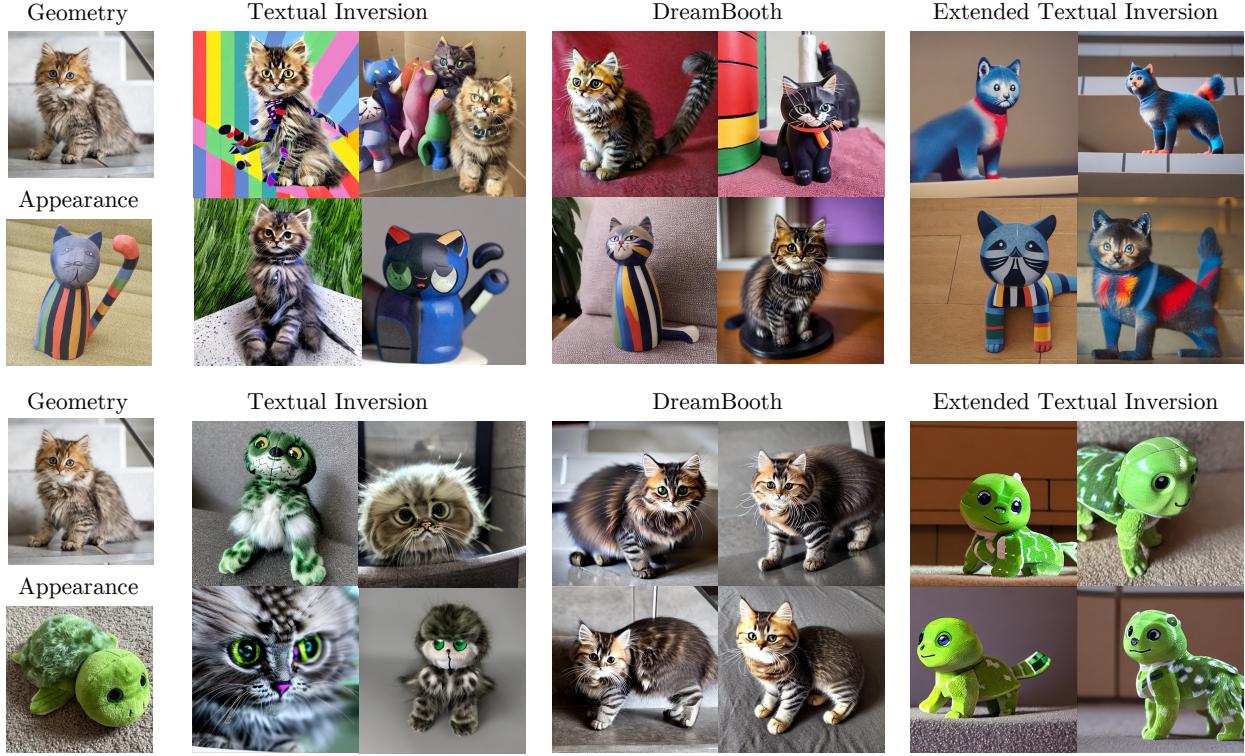


Figure 14. **Style Mixing comparison.** We compare against Textual Inversion [15] and Dreambooth [45] baselines. For TI we independently invert target subject and target appearance, and generate the images with the sentence ”`<object>` that looks like `<appearance>`”. Prompt variations did not make any remarkable improvements. The style source concept was inverted with style prompts (see [15] for details). DreamBooth was trained with a pair of subjects. Our proposed Extended Textual Inversion clearly outperforms both baselines.

## References

- [1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019. [2](#)
- [2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8296–8305, 2020. [2](#)
- [3] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit H. Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing, 2021. [2](#)
- [4] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *arXiv preprint arXiv:2206.02779*, 2022. [2](#)
- [5] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18208–18218, 2022. [2](#)
- [6] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kassten, and Tali Dekel. Text2live: Text-driven layered image and video editing. *arXiv preprint arXiv:2204.02491*, 2022. [2](#)
- [7] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 2020. [2](#)
- [8] Amit H Bermano, Rinon Gal, Yuval Alaluf, Ron Mokady, Yotam Nitzan, Omer Tov, Oren Patashnik, and Daniel Cohen-Or. State-of-the-art in the architecture, methods and applications of stylegan. In *Computer Graphics Forum*, volume 41, pages 591–611. Wiley Online Library, 2022. [2](#)
- [9] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023. [2](#)
- [10] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *CoRR*, abs/2104.14294, 2021. [6](#)
- [11] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023. [2](#)
- [12] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022. [2](#)

- [13] Antonia Creswell and Anil Anthony Bharath. Inverting the generator of a generative adversarial network. *IEEE transactions on neural networks and learning systems*, 30(7):1967–1974, 2018. 2
- [14] Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. Vqgan-clip: Open domain image generation and editing with natural language guidance. *arXiv preprint arXiv:2204.08583*, 2022. 2
- [15] 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. 2, 4, 6, 9, 10, 16
- [16] Rinon Gal, Moab Arar, Yuval Atzmon, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Designing an encoder for fast personalization of text-to-image models. *arXiv preprint arXiv:2302.12228*, 2023. 9
- [17] 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. 2
- [18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2
- [19] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 13
- [20] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. 6
- [21] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Proc. NeurIPS*, 2020. 2
- [22] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 2
- [23] 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. 2
- [24] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Hui-Tang Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. *ArXiv*, abs/2210.09276, 2022. 2
- [25] Gwanghyun Kim and Jong Chul Ye. Diffusionclip: Text-guided image manipulation using diffusion models. *arXiv preprint arXiv:2110.02711*, 2021. 2
- [26] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. *CVPR*, 2023. 2, 6, 16
- [27] Gihyun Kwon and Jong Chul Ye. Clipstyler: Image style transfer with a single text condition. *arXiv preprint arXiv:2112.00374*, 2021. 2
- [28] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. *arXiv preprint arXiv:2211.10440*, 2022. 2
- [29] Zachary C Lipton and Subarna Tripathi. Precise recovery of latent vectors from generative adversarial networks. *arXiv preprint arXiv:1702.04782*, 2017. 2
- [30] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds, 2022. 13
- [31] Chenlin Meng, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. 2
- [32] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. *arXiv preprint arXiv:2211.09794*, 2022. 2
- [33] 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. 2
- [34] Yotam Nitzan, Kfir Aberman, Qiupei He, Orly Liba, Michal Yarom, Yossi Gandelsman, Inbar Mosseri, Yael Pritch, and Daniel Cohen-Or. Mystyle: A personalized generative prior. *ACM Transactions on Graphics (TOG)*, 41(6):1–10, 2022. 2
- [35] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. *arXiv preprint arXiv:2302.03027*, 2023. 2
- [36] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962. 8
- [37] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. *arXiv preprint arXiv:2103.17249*, 2021. 2
- [38] 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. *arXiv preprint arXiv:2103.00020*, 2021. 4
- [39] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2
- [40] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. *arXiv preprint arXiv:2008.00951*, 2020. 2
- [41] Daniel Roich, Ron Mokady, Amit H. Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Transactions on Graphics (TOG)*, 2022. 2
- [42] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 2

- [43] 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 (CVPR)*, pages 10684–10695, June 2022. 3, 4, 6
- [44] Murray Rosenblatt. Remarks on some nonparametric estimates of a density function. *The annals of mathematical statistics*, pages 832–837, 1956. 8
- [45] 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. 2, 6, 9, 10
- [46] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 2
- [47] 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. 2
- [48] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020. 2
- [49] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019. 2
- [50] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *arXiv preprint arXiv:2102.02766*, 2021. 2
- [51] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. *arXiv preprint arXiv:2211.12572*, 2022. 2
- [52] Dani Valevski, Matan Kalman, Yossi Matias, and Yaniv Leviathan. Unitine: Text-driven image editing by fine tuning an image generation model on a single image. *arXiv preprint arXiv:2210.09477*, 2022. 2
- [53] Andrey Voynov and Artem Babenko. Rpgan: random paths as a latent space for gan interpretability. 2
- [54] Su Wang, Chitwan Saharia, Ceslee Montgomery, Jordi Pont-Tuset, Shai Noy, Stefano Pellegrini, Yasumasa Onoe, Sarah Laszlo, David J Fleet, Radu Soricu, et al. Imagen editor and editbench: Advancing and evaluating text-guided image inpainting. *arXiv preprint arXiv:2212.06909*, 2022. 2
- [55] Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. High-fidelity gan inversion for image attribute editing. *ArXiv*, abs/2109.06590, 2021. 2
- [56] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12863–12872, 2021. 2
- [57] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. Gan inversion: A survey, 2021. 2
- [58] Raymond A. Yeh, Chen Chen, Teck Yian Lim, Alexander G. Schwing, Mark Hasegawa-Johnson, and Minh N. Do. Semantic image inpainting with deep generative models, 2017. 2
- [59] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision-ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014. 2
- [60] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *European conference on computer vision*, pages 597–613. Springer, 2016. 2

## Supplementary

### 7.1. Regularization

When applying style mixing, we discovered that optimizing XTI with an additional density regularization loss term (Equation 2) improves the mixing capability while maintaining the overall quality of the inversion. To achieve this, we use an extra loss term  $-\lambda \cdot \sum_{i=1}^l \log p_{\mathcal{E}}(e_i)$ , where  $\lambda = 0.002$  serves as a small regularization scale. This loss term encourages the newly added look-up table embedding to be even more regular, causing the optimized token distribution from Figure 11 to shift closer to the original token distribution. We contend that this is particularly advantageous for the mixture application because in this scenario, generation is conditioned by two different XTI tokens, making it crucial to have them interact naturally, i.e., with the two tokens lying closer to the natural language manifold.

However, applying this regularization term to the original TI for subject recontextualization leads to a degradation in subject similarity of the inversion. Even with a small  $\lambda$  scaling factor, this regularization enforces significant simplification in the inversion process, leaving the reconstructed token with limited freedom and expressivity. Meanwhile, although the drop in quality for XTI is minimal when the regularization is added, we do not use it by default for the recontextualization task for XTI because it would increase its complexity (another hyperparameter) and convergence rate.

## 8. Further Results and Details

Figures 15 and 16 provide more examples of geometry and style mixing. In Figure 15 objects prompts are passed to a wider range of layers compared to Figure 16, enforcing higher source geometry alignment.

Figure 17 provides more uncurated examples generated with Textual Inversion and Extended Textual Inversion.

As for inference we use the PNDM scheduler [30] with 50 denoising steps and a classifier-free guidance [19] scale of 7.5.

We implemented DreamBooth with a learning rate of 5e-6, a batch size of 4, and conducted 400 finetuning steps per-subject, using Stable Diffusion. Our optimization focused solely on the U-net weights and did not involve prior preservation.

### 8.1. Coarse/Fine Layers Split

We provide the details of the layer subsets we used in the shape-style mixing experiments in Figures 5 and 7. We name the cross-attention layers of Stable Diffusion U-net as follows, in the order they appear in the U-net:

(64, 'down', 0), (64, 'down', 1),  
(32, 'down', 0), (32, 'down', 1),

(16, 'down', 0), (16, 'down', 1),  
(8, 'down', 0), (16, 'up', 0),  
(16, 'up', 1), (16, 'up', 2),  
(32, 'up', 0), (32, 'up', 1),  
(32, 'up', 2), (64, 'up', 0),  
(64, 'up', 1), (64, 'up', 2).

The first number represents the spatial resolution, 'down' / 'up' represents whether the layer is on the downward (contracting) or upward (expansive) part of the U-net, and the third number indicates the index among the cross-attention layers of the same resolution and direction.

In Figure 7 we use the following growing sequence of cross-attention layer subsets:

- 0: Empty set
- 1: Layer (8, 'down', 0) only
- 2: (16, 'down', 1) - (8, 'down', 0)
- 3: (16, 'down', 1) - (16, 'up', 0)
- 4: (16, 'down', 0) - (16, 'up', 0)
- 5: (16, 'down', 0) - (16, 'up', 1)
- 6: (16, 'down', 0) - (16, 'up', 2)
- 7: (64, 'down', 0) - (64, 'up', 2)

The ranges listed above are inclusive.

In Figures 2 and 11 in the main text we condition the layers (8, 'down', 0), (16, 'up', 0) on the target shape textual embeddings, and the other layers on the target style textual embeddings. In Figure 3 in the main text we provide the target shape textual embeddings to layers (16, 'down', 1) - (16, 'up', 0).

### 8.2. Text Prompts

#### 8.2.1 Cross-Attention Analysis (Fig. 5)

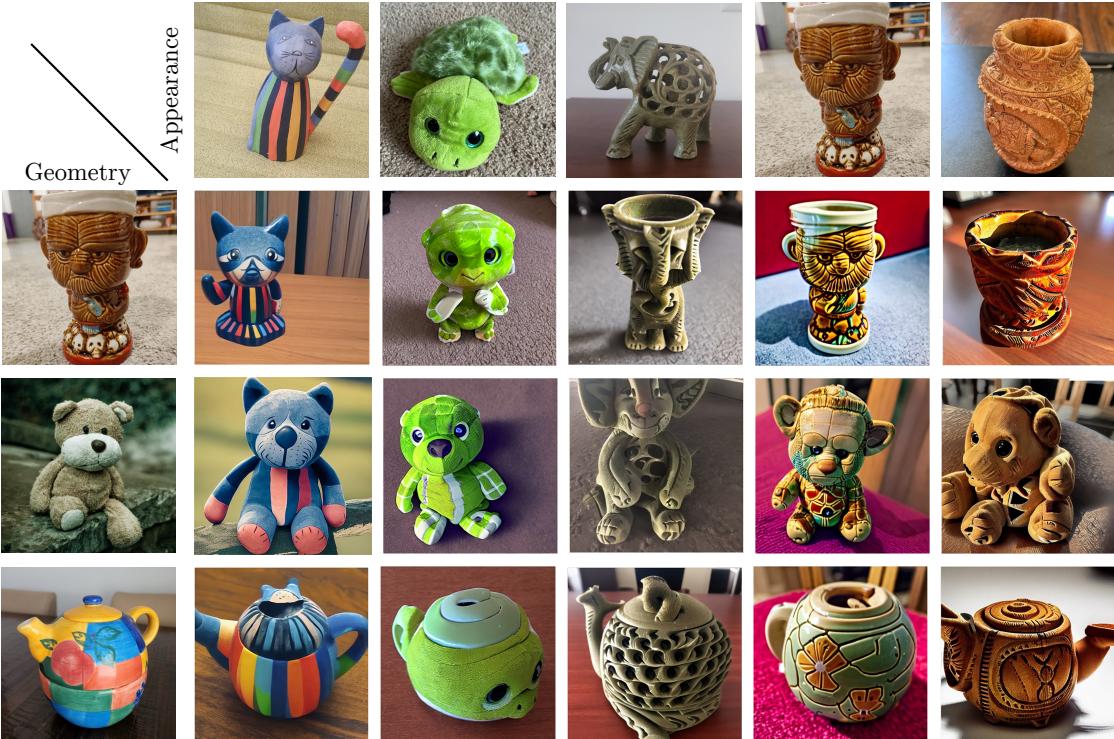
We used the following lists of objects and appearances for generating the "appearance object" and "object, appearance" prompts in Figure 5:

Objects (50): "dog", "cat", "tree", "chair", "book", "phone", "car", "bike", "lamp", "table", "flower", "desk", "computer", "pen", "pencil", "lamp", "television", "picture", "mirror", "shoe", "boot", "sandals", "house", "building", "street", "park", "river", "ocean", "lake", "mountain", "chair", "couch", "armchair", "bookcase", "rug", "lampshade", "fan", "conditioner", "heater", "door", "window", "bed", "pillow", "blanket", "curtains", "kitchen", "refrigerator", "stove", "oven", "microwave"

Appearances (20): "fuzzy", "shiny", "bright", "fluffy", "sparkly", "dull", "smooth", "rough", "jagged", "striped",



**Figure 15. Style mixing examples.** *Top row:* Style source subjects, *First column:* Geometry source subjects. The geometry subject’s tokens are passed to the three layers in the range  $(16, \text{'down'}, 1) - (16, \text{'up'}, 0)$ , while all the rest are conditioned on the appearance subject’s token.



**Figure 16. Style mixing with more appearance layers.** *Top row:* Style source subjects, *First column:* Geometry source subjects. Here the geometry subject’s tokens are passed only to two layers  $(8, \text{'down'}, 0)$  and  $(16, \text{'up'}, 0)$ . Thus this emphasizes the appearance subject’s token more, resulting in a more dominant appearance compared to the previous setup in Figure 15.

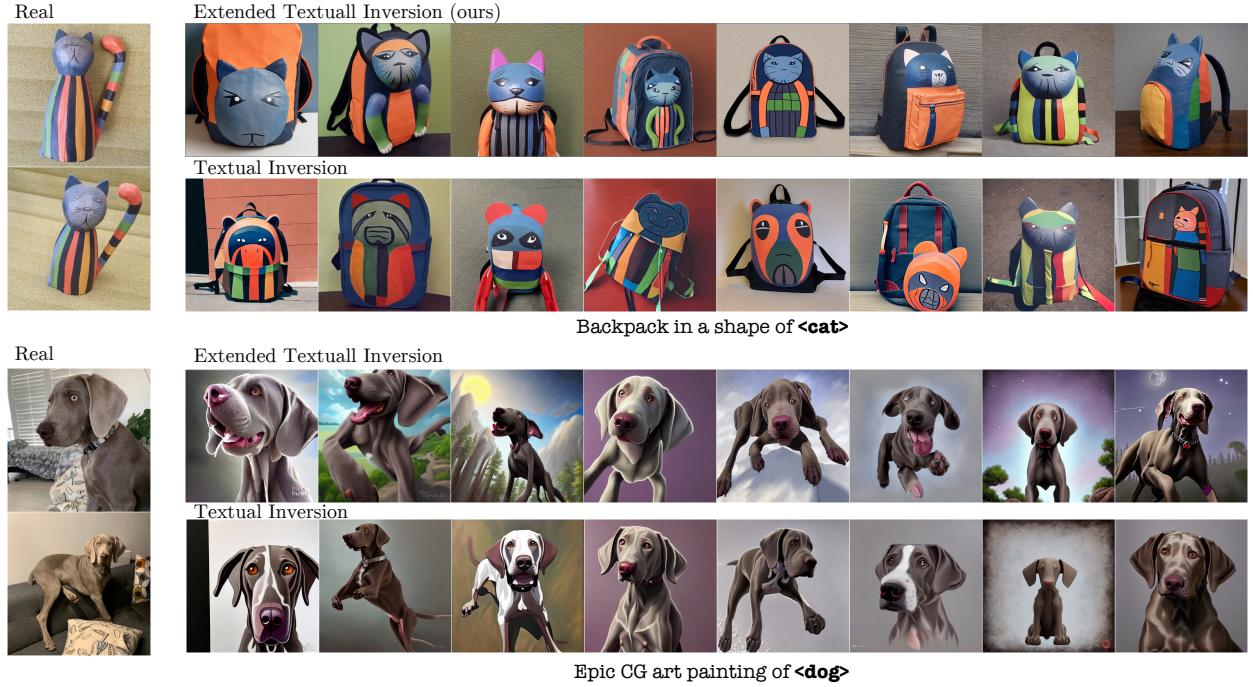


Figure 17. Samples generated with Textual Inversion (TI) and the proposed Extended Textual Inversion (XTI). XTI has better text alignment while providing more accurate subject reconstruction.

"painting", "retro", "vintage",  
 "modern", "bohemian", "industrial",  
 "rustic", "classic", "contemporary",  
 "futuristic"

### 8.2.2 Image Attributes Analysis (Fig. 7)

For Figure 7, we used the following object, color and style words to generate the prompts:

Objects (13): "chair", "dog", "book",  
 "elephant", "guitar", "pillow",  
 "rabbit", "umbrella", "yacht", "house",  
 "cube", "sphere", "car"

Colors (11): "black", "blue", "brown",  
 "gray", "green", "orange", "pink",  
 "purple", "red", "white", "yellow"

Style descriptions (7): "watercolor", "oil  
 painting", "vector art", "pop art  
 style", "3D rendering", "impressionism  
 picture", "graffiti"

### 8.2.3 Text Similarity Metric Prompts

For Text Similarity evaluation we use the following 14 prompts:

"A photograph of <token>", "A photo  
 of <token> in the jungles", "A photo

of <token> on a beach", "Aquarelle  
 painting of <token>", "Oil painting  
 of <token>", "Marc Chagall painting of  
 <token>", "Sketch drawing of <token>",  
 "Night photograph of <token>",  
 "Professional studio photograph of  
 <token>", "3d rendering of <token>",  
 "Fantasy CG art painting of <token>",  
 "A statue of <token>", "A photograph of  
 two <token> on a table", "App icon of  
 <token>".

Here <token> represents the placeholder for XTI inversion tokens to be replaced with the corresponding textual description in Table 2.

### Original Dataset

	Text Description
elephant	a statue of an elephant
cat_statue	a statue of a cat
colorful_teapot	a colorful teapot
clock	an alarm clock
mug_skulls	a cup with a mummy
physics_mug	a black cup with math equations
red_teapot	a red teapot
round_bird	a round bird sculpture
thin_bird	a sculpture of a thin bird
barn	an old wooden barn
cat	a kitten
dog	a grey dog
teddybear	a teddy bear
tortoise_plushy	a tortoise plush
wooden_pot	an artistic wooden pot

Table 2. Detailed text descriptions for each dataset. The first 9 correspond to the datasets provided in [15], and the remaining 6 correspond to the datasets provided in [26].

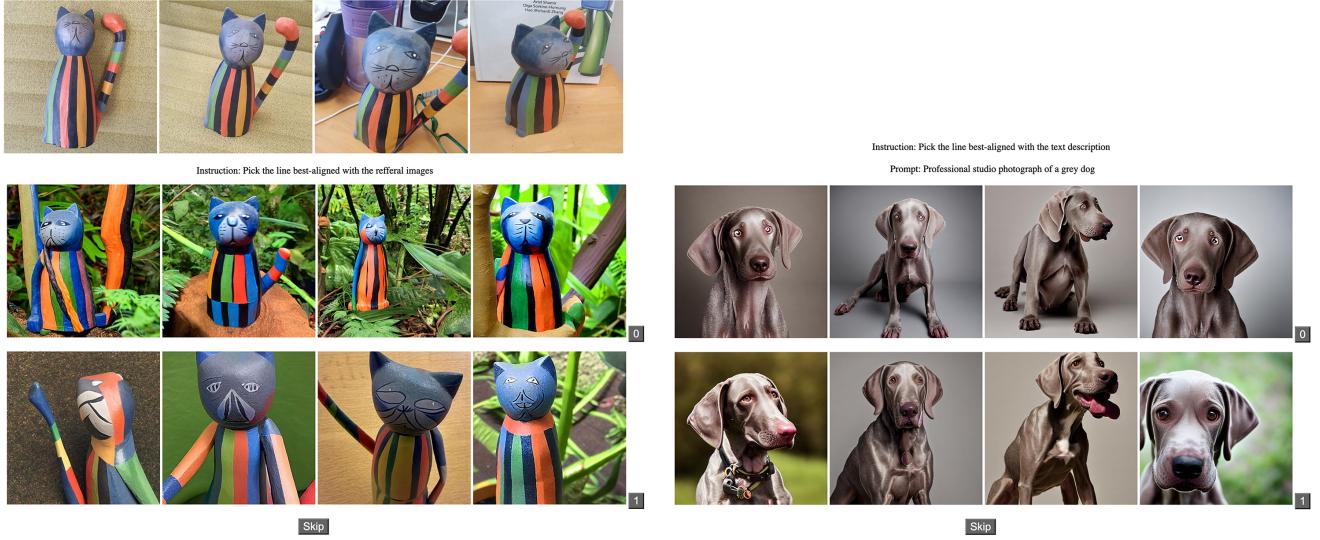


Figure 18. Human labeling interface. On the *left* we depict a sample task to evaluate subject similarity, and on the *right* the task to evaluate text similarity. The comparing methods raws are always shuffled. Both methods use the same random seed.