An Image is Worth Multiple Words: Multi-attribute Inversion for Constrained Text-to-Image Synthesis (Supplementary material)

Aishwarya Agarwal, Srikrishna Karanam, Tripti Shukla, and Balaji Vasan Srinivasan
Adobe Research, Bengaluru India
{aishagar,skaranam,trshukla,balsrini}@adobe.com

1. Appendix

In Section 1.1, we show additional results for joint layer-timestep analysis to show further evidence for how certain attributes can be disentangled when both layer and timestep dimension are considered jointly, which otherwise can not be disentangled across a single dimension (as in P+ [11] and ProSpect [13]). In Section 1.2, we show quantitative analysis results for style and object attributes that confirm the layer and timestep partitioning of Section 3.2 in the main paper. In Section 1.3, we show more qualitative results for comparing MATTE with baselines. Here we also explain in detail how the prompt conditionings for the baselines P+ and ProSpect are computed. In Section 1.4, we talk about the implementation details and the images used for evaluation. In Section 1.5, we provide more details on the quantitative evaluation setup followed for reporting the results comparing MATTE with baselines in Table 2 in the main paper. Finally, we conclude with some discussion on limitations of our method in Section 1.7.

1.1. Additional Layer-timestep Analysis

As discussed in the main paper, attributes like `color` and `layout` that are captured along the same timesteps (from Prospect [13]) can be disentangled along the layer dimension. Similarly, geometric attributes like `object` and `layout` that are captured along the same layers (from P+ [11]) can be disentangled along the timestep dimension. We show additional qualitative results to demonstrate the conclusions stated above.

Consider Figure 1 for an example on layout-color disentanglement using joint layer-timestep prompt conditionings. In Figure 1(a), (b) and (c), one can note that we get a `blue` ball despite the color being specified as `red` in the `coarse` layers. In Figure 1(a) and (b), we get a ball placed `on a table` and `under a table` respectively as expected, because the corresponding layout conditionings were given as input to all U-Net [10] layers. In Figure 1(c), we notice that we get a ball `on a table` despite specifying `under a table` in the `moderate` layers. This clearly indicates that the `coarse` layers are dominantly responsible for determining the layout. To summarise, this example shows that while `color` and `layout` are captured along the same timesteps, they can be disentangled along the layer dimension (L_3 - L_5 & L_{10} - L_{13} for color and L_6 - L_9 for layout).

Similarly, consider the example in Figure 2, where we show `object` and `layout` that are captured along the same layers can be disentangled along the timestep dimension. Here, based on the final generated image (a standing cow), one can note that the text conditions corresponding to the object `cow` were specified in stage t_2, t_3 and had the most impact on the final image. For instance, despite the input `cat` in t_1 stage, the final image has a cow. Similarly, despite specifying the layout `sitting` in stages t_2, t_3, the final generation only respected `standing` that was provided in stage t_1. This shows that while `object` and `layout` are captured along the same layers, they can be disentangled along the timestep dimension (t_2, t_3 for object and t_1 for layout).

Before we move on to the next example, we had summarised from our analysis in the main paper that fine layers (L_1 - L_2 & L_{14} - L_{16}) and the stage t_4 have no impact on any of the four attributes. `Color` and `style` are both captured in the initial denoising stages (t_1, t_2) and across moderate U-Net layers (L_3 - L_5 & L_{10} - L_{13}). `Object` semantics are captured along the middle denoising stages (t_2, t_3) and across the coarse U-Net layers (L_6 - L_9). Finally, `layout` is captured in the very initial denoising stage (t_1) across coarse layers (L_6 - L_9).

On that note, we show another example in Figure 3, similar to the joint multi-prompt conditioning example presented in Figure 5 in the main paper, which demonstrates the properties summarised above. For the final generated image (*a blue ball placed on a table*), one can note that the textual conditionings corresponding to each of key attributes in the prompt (`blue`, `ball`, and `on the table`) were specified only across a subset of layers and only along specific

timesteps. These observations are consistent with the analysis we summarised for each of the attributes. For instance, , despite specifying `white` in L_1 - L_2 & L_{14} - L_{16} layers, and color `red` in (L_6 - L_9) layers, we still see a `blue` colored ball based on the color specified in moderate (L_3 - L_5 & L_{10} - L_{13}) layers. Similarly, the layout is captured in the initial stages and across the coarse set of layers. We also show per-layer attention maps across denoising timesteps in Figure 4 which confirms that layout is predominantly captured in layers L_6, L_8, and L_9.

1.2. Quantitative Layer-timestep Analysis

We continue the discussion on joint layer-timestep quantitative analysis as in Section 3.2 in the main paper. The `style` attribute follows the same patterns as that of the color as shown in the main paper. We discuss the `object` attribute in detail, the results for which are plotted in Figure 5. Consider a particular y-label in the plot shown for p_1 (recall that as we move from bottom to top along the y-axis, an increasing number of layers starting from the coarse ones are replaced with p_2 conditioning instead of p_1). One can observe that the drop in similarity scores is maximum initially when the first few layers from the coarse set are removed (along y-axis). On the other hand, when we consider the timestep conditioning effect, the scores in the initial stages of replacement are unaffected ($p1/p2$ replaced by $p3/p4$ progressively) as can be noted from the width of the horizontal bars. But after a few timestep stages, the width of the horizontal plots start reducing, thereby confirming that the object attribute is captured in the middle denoising stages.

1.3. Additional Qualitative Results and Setup Details

We show additional results comparing MATTE with the closest baselines ProSpect [13] and P+ [11]. We first explain how we generate images using P+ and ProSpect given a prompt with an example. Consider the example shown in column 1 in Figure 6. The goal here is to generate images of a `dog` in `oil painting` style following the `color` properties of the reference image. The first step here is to run the inversion algorithms of P+ and ProSpect for the reference image, and get a set of textual conditionings $< x_i >$ where $i = 1, \cdots, 16$ for P+, and $< y_j >$ where $j = 1, \cdots, 10$ for ProSpect. Next, depending upon the attributes we want to transfer from the reference image (`color` here), we retain the textual conditionings learned during inversion in P+ and Prospect as part of the final conditionings used as input along the 16 layers and 10 timestep stages respectively. The decision of retaining conditionings is made on the basis of which set of timesteps/layers are important for capturing the attribute of interest (color here). Since we know color is captured across the shallow U-Net layers in P+, and across the initial denoising stages in ProSpect, the prompt that goes as input to P+ across the 16 U-Net layers is:

[$< x_1 >$ dog in oil painting style,
$< x_2 >$ dog in oil painting style,
$< x_3 >$ dog in oil painting style,
$< x_4 >$ dog in oil painting style,
$< x_5 >$ dog in oil painting style,
dog,
dog,
dog,
dog,
$< x_{10} >$ dog in oil painting style,
$< x_{11} >$ dog in oil painting style,
$< x_{12} >$ dog in oil painting style,
$< x_{13} >$ dog in oil painting style,
$< x_{14} >$ dog in oil painting style,
$< x_{15} >$ dog in oil painting style,
$< x_{16} >$ dog in oil painting style].

Similarly, the prompt for ProSpect across the 10 denoinsing timestep stages is:

[$< y_1 >$ dog in oil painting style,
$< y_2 >$ dog in oil painting style,
$< y_3 >$ dog in oil painting style,
$< y_4 >$ dog in oil painting style,
dog in oil painting style,
dog in oil painting style,
dog in oil painting style,
dog in oil painting style,
dog in oil painting style,
dog in oil painting style].

We next discuss the results comparing MATTE with P+ and ProSpect in Figure 6. Consider the example in column 1. Here the goal is to generate a `dog` in `oil painting style` while retaining only the `color` properties from the reference image. We see that MATTE captures everthing (dog, oil painting style and color attribute from reference image) accurately. While in ProSpect, even though the colors got transferred from the reference image, but it has generated dogs following the layout of the inkpot shown in the reference image. This is because, as seen previously, layout and color are captured across similar denoising timesteps, hence disentangling the two is not possible in ProSpect (as inversion in ProSpect is across timestep dimension only). Similarly, for P+, we see that the generated dogs follow the oil painting style but are unable to capture the color of the inkpot. This again is because color and style are captured in same layers in P+, so either color and style both get transferred together or none gets transferred. One can make similar observations across the examples shown in other columns too which clearly indicated that MATTE is able to constrain the generation of images on attributes from

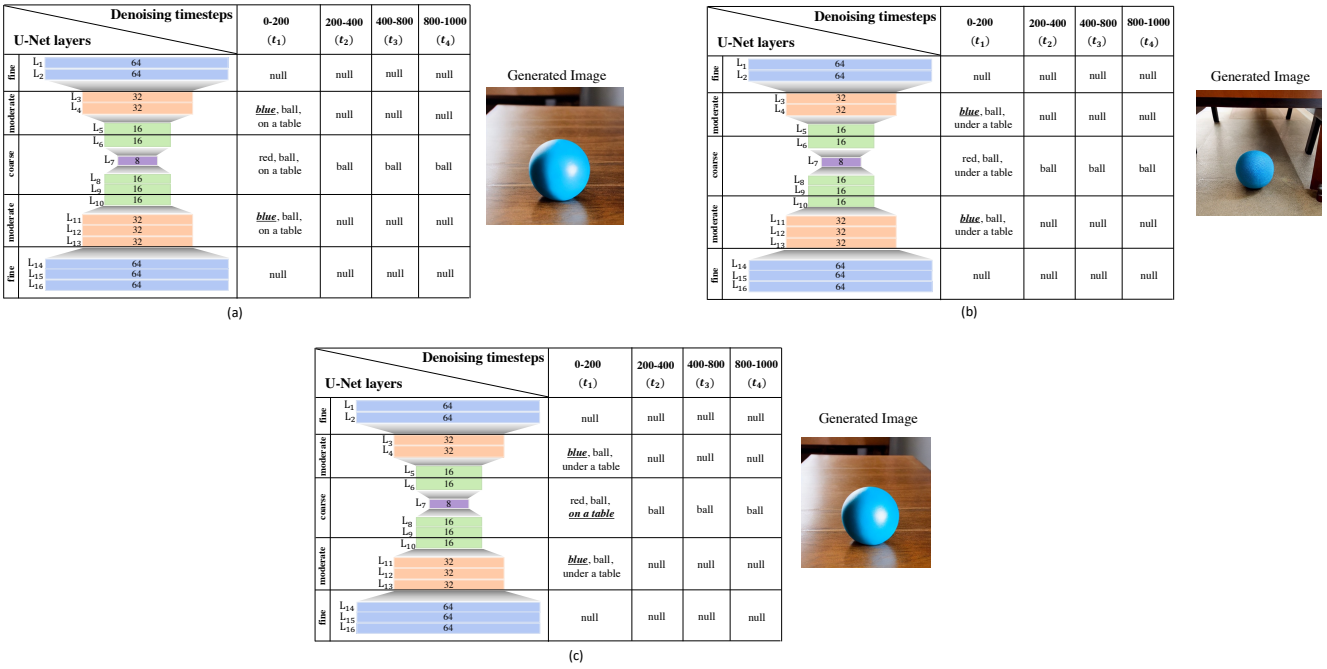


Figure 1. Layout-Color disentanglement.

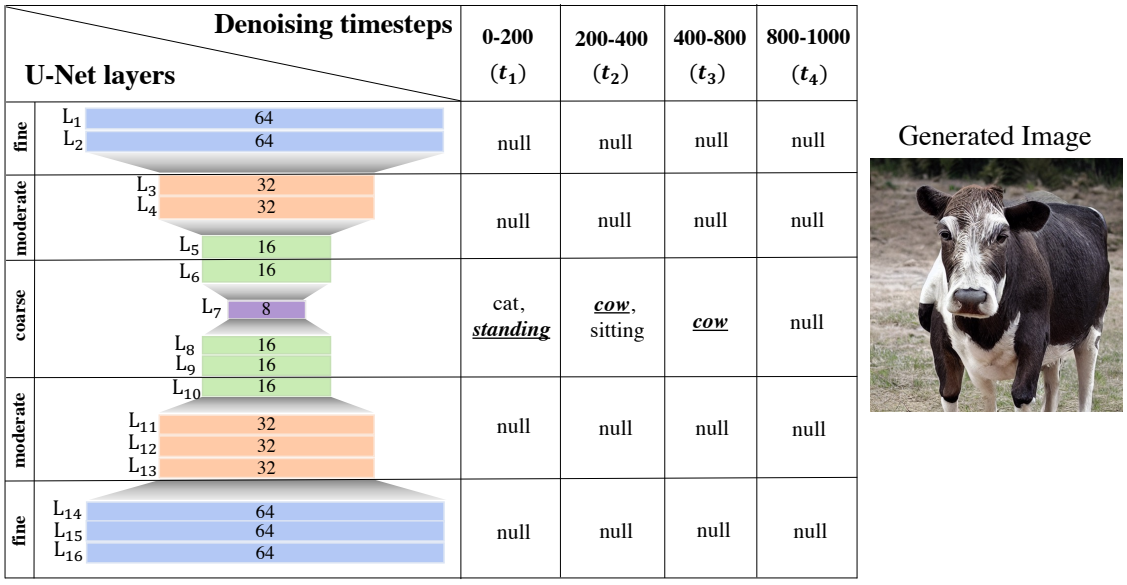


Figure 2. Layout-Object disentanglement.

reference image in a disentangled fashion much better than the closest baselines.

We also show additional comparison results of MATTE with both diffusion-specific [2, 7, 12] as well as conventional baselines [8] for style transfer in Figure 7 where one can observe the MATTE substantially outperforms the baselines.

1.4. Implementation Details and Dataset

We follow the same set of styles, objects and colors as described in P+ [11] for all our evaluations and trainings.

Specifically, during MATTE inversion technique (Section 3.2 in the main paper), the set of styles used to randomly choose styles from was:

```
["oil painting", "vector art", "pop art
```

U-Net layers \ Denoising timesteps		0-200 (t_1)	200-400 (t_2)	400-800 (t_3)	800-1000 (t_4)
fine	L_1 64, L_2 64	white, lizard, under a table	white, lizard, under a table	white, lizard, under a table	null
moderate	L_3 32, L_4 32, L_5 16	***blue***, cat, under a table	***blue***, cat, under a table	green, cat, under a table	null
coarse	L_6 16, L_7 8, L_8 16, L_9 16	red, cat, ***on a table***	red, ***ball***, under a table	red, ***ball***, under a table	null
moderate	L_{10} 16, L_{11} 32, L_{12} 32, L_{13} 32	***blue***, cat, under a table	***blue***, cat, under a table	green, cat, under a table	null
fine	L_{14} 64, L_{15} 64, L_{16} 64	white, lizard, under a table	white, lizard, under a table	white, lizard, under a table	null

Generated Image

Figure 3. Multi-prompt conditioning across U-Net layers and denoising timesteps jointly.

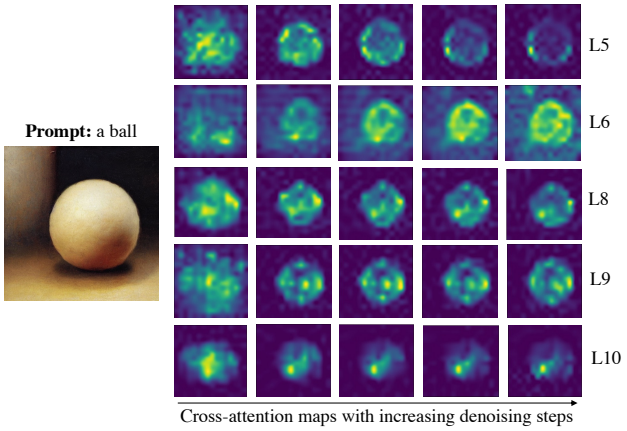
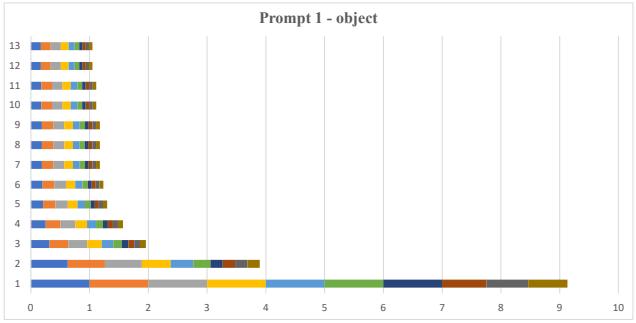
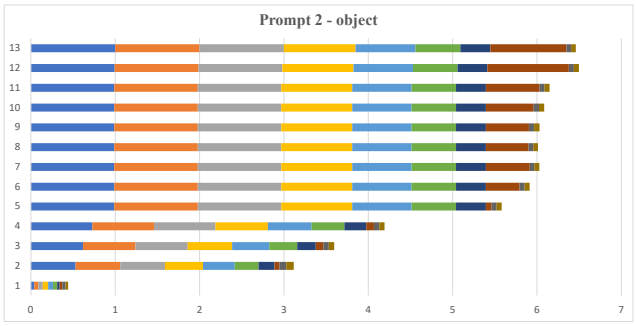


Figure 4. Cross-attention maps for analysing layout.



(a) Prompt 1



(b) Prompt 2

Figure 5. The figure demonstrates the similarity scores obtained from our joint layer-timestep analysis for understanding which layers/timesteps object is captured in. This is a combination across all 13 layer conditionings for the analysis shown in in Figure 6 in the main paper.

```
style",
"3D rendering", "impressionism picture",
"graffiti", "fuzzy",
"shiny", "bright", "fluffy", "sparkly",
"dull", "smooth",
"rough", "jagged", "striped",
"painting", "retro", "vintage",
"modern", "bohemian", "industrial",
"rustic", "classic",
"contemporary", "futuristic"]
```

For the quantitative evaluations in Section 4 in the main paper, we use the following sets of objects, style and colors (again from P+ [11]):

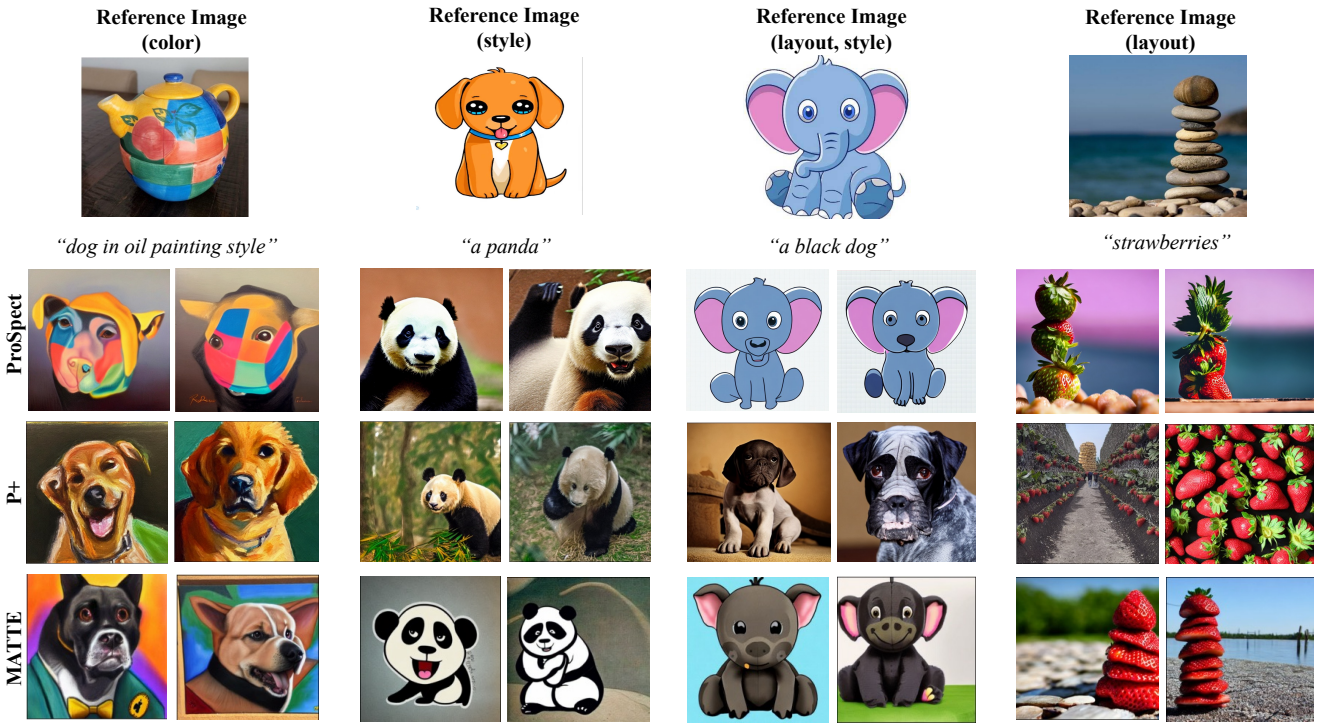


Figure 6. Comparison of MATTE with recent state-of-the-art methods for reference-constrained text-to-image generation.

Objects = ["chair", "dog", "book", "elephant", "guitar", "pillow", "rabbit", "umbrella", "yacht", "house", "cube", "sphere", "car'"]

Colors = ["black", "blue", "brown", "gray", "green", "orange", "pink", "purple", "red", "white", "yellow"]

Styles = ["watercolor", "oil painting", "vector art", "pop art style", "3D rendering", "impressionism picture", "graffiti"]

Finally, we show the images used for different evaluation setups in Figure 8.

Note that all our experiments were conducted using a single A10G GPU with a batch size of 2.

1.5. Quantitative Evaluation Setup Details

We presented an evaluation to quantify the disentanglement of different pairs of attributes in the main paper in Table 2, Section 4. Here, we explain the details of how we compute the CLIP Image-text similarities reported in the paper. We use the set of images shown in Figure 8 and the set of attributes discussed in Section 1.4. For each reference image, our goal was to evaluate the inversion techniques in aspects of (i) preserving/transferring an attribute from the reference image and (ii) generating images following attributes mentioned in the text prompt. We considered 6 unique pairs of attributes for this comparison namely `layout-color`, `layout-object`, `layout-style`, `color-object`, `color-style` and `object-style`. Consider the case of `color-object` disentanglement evaluation in the context of reference image based attribute-aware text-to-image generation. The attribute mentioned first (`color` here) is the one to be transferred from the reference image, whereas the latter (`object` here) comes from the text prompt. For each of the baselines P+ and ProSpect, we generate final prompt conditionings in the same fashion as explained in Section 1.3 by retaining the textual conditionings responsible for capturing the attribute to be transferred from reference image (color here). For the attribute that comes from the text prompt (objects here), we iterate over a set of different objects following the list of objects mentioned in Section 1.4 and generate a set of 64 images for each color-object pair. We then compute CLIP Image-text similarities between the generated images and the ground truth object used to generate those images. Similarly, we also compute CLIP Image-text similarities between the

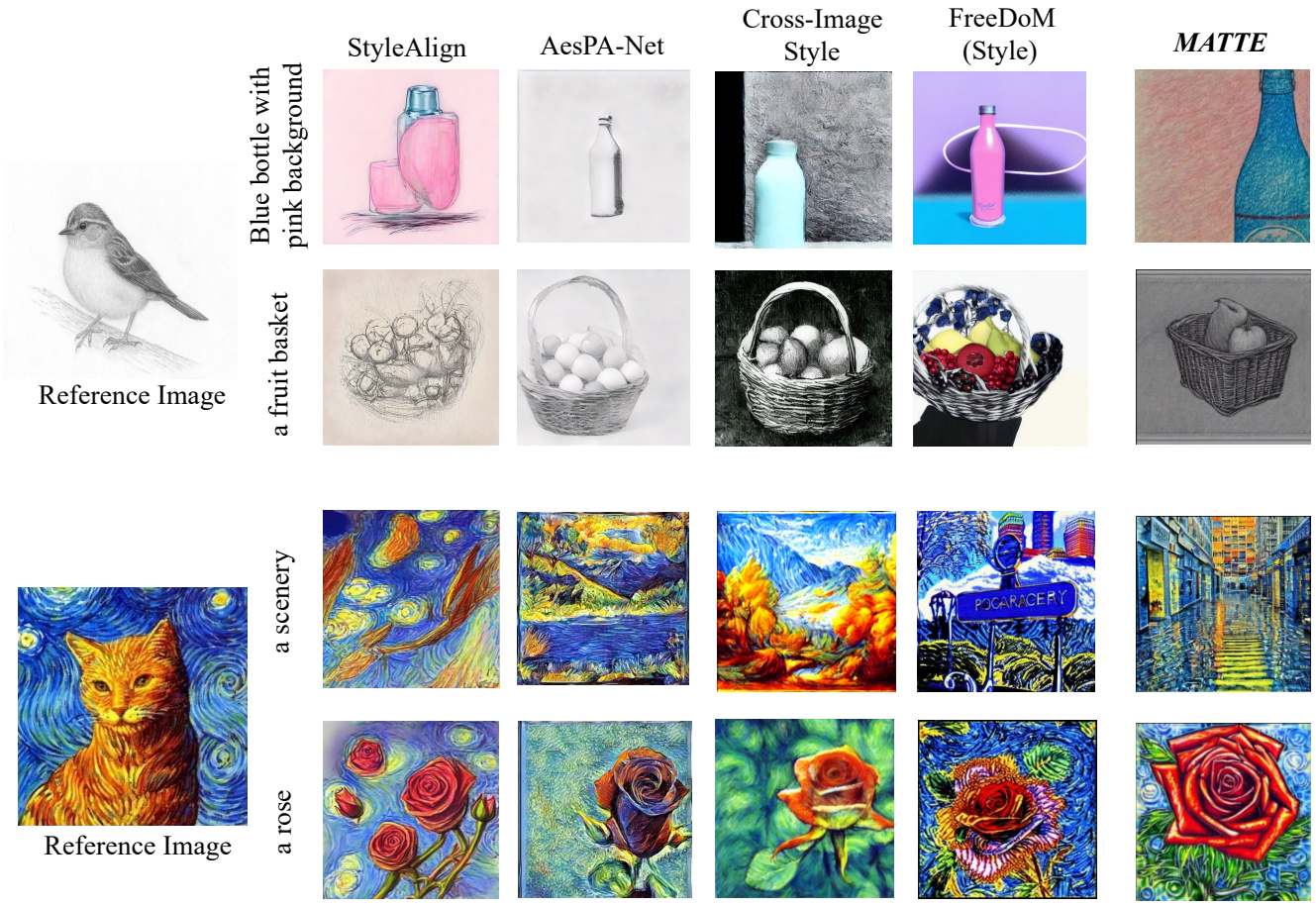


Figure 7. Comparison of MATTE with recent state-of-the-art methods for style transfer.

generated images and the corresponding ground truth for the attribute to be transferred from reference image wherever possible, followed by an averaging of the two similarities (for color in color-object case, ground truth colors are extracted from the reference image using Color Thief [4]). Similarly, these CLIP based Image-text similarities are computed for other attribute pairs for MATTE and the closest baselines P+ and ProSpect, results of which are reported in Table 2 in the main paper.

1.6. User Study

We conduct a user study with the generated images where we ask survey respondents to select which set of images (among sets from three different methods, see Table 1) best represents the input constraints. The user is presented with a reference image, a text prompt, and a set of attributes from the reference image that should ideally get transferred to the final generated image (see Figure **??** for an example). From Table 1, our method's results are preferred by a majority of the survey respondents, thus providing additional evidence for the impact of our proposed inversion technique in constraining text-to-image generation on different attributes of reference images in a disentangled fashion.

Method	Percentage
P+ [11]	12.2%
ProSpect [13]	13.2%
MATTE	**74.6%**

Table 1. Results from a user survey with 24 respondents.

1.7. Limitations

In this Section, we briefly discuss a few limitations of MATTE when seen in a constrained text-to-image generation setup. Firstly, the optimization of the embeddings learned for the four tokens namely $< c >$, $< l >$, $< o >$ and $< s >$ during inversion is a slow process (MATTE converges

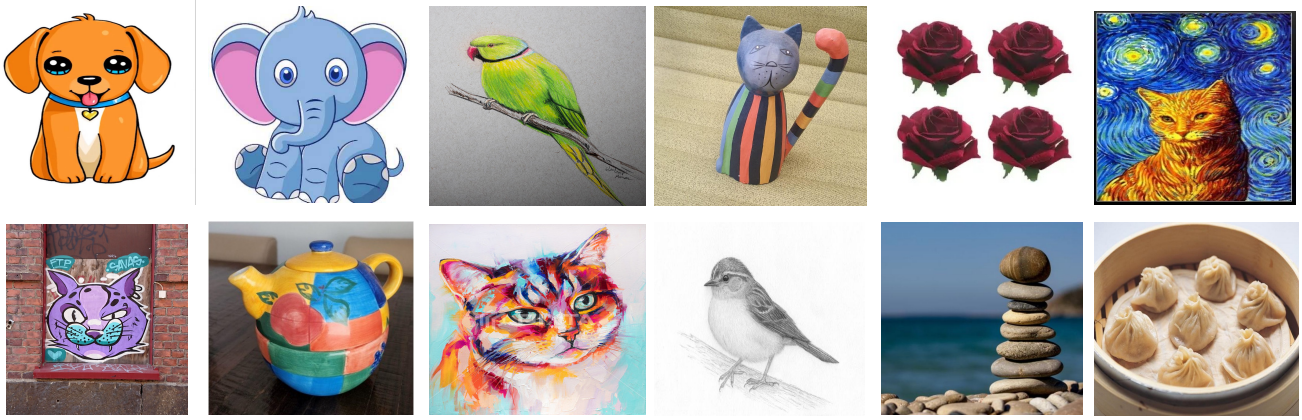


Figure 8. Images used for evaluation.

faster than TI [6], but is still slow), thereby posing a limitation to its' practical applicability. Secondly, since MATTE doesn't involve fine-tuning model weights, the final constrained text-to-image generation pipeline after MATTE inverts the reference image into disentangled tokens is limited by the generation abilities of the base diffusion model. For instance, omission of objects mentioned in the text prompt is a known limitation of diffusion models [1, 3, 5, 9]. So, given a prompt "`a <c> colored cat playing with a dog`" (where $< c >$ is extracted from a reference image using MATTE) to the base diffusion model, MATTE will ensure that the cat generated is $< c >$ colored but MATTE can not enforce the presence of a cat in the final generated image. Moreover, since we are building on top of existing text-to-image models, any potential fairness considerations for these base models will flow to our method as well.

References

[1] Aishwarya Agarwal, Srikrishna Karanam, KJ Joseph, Apoorv Saxena, Koustava Goswami, and Balaji Vasan Srinivasan. A-star: Test-time attention segregation and retention for text-to-image synthesis. *arXiv preprint arXiv:2306.14544*, 2023.

[2] Yuval Alaluf, Daniel Garibi, Or Patashnik, Hadar Averbuch-Elor, and Daniel Cohen-Or. Cross-image attention for zero-shot appearance transfer. *arXiv preprint arXiv:2311.03335*, 2023.

[3] 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. *arXiv preprint arXiv:2301.13826*, 2023.

[4] Lokesh Dhakar. Color thief. *Retrieved*, 2015.

[5] Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. Training-free structured diffusion guidance for compositional text-to-image synthesis. *arXiv preprint arXiv:2212.05032*, 2022.

[6] 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.

[7] Amir Hertz, Andrey Voynov, Shlomi Fruchter, and Daniel Cohen-Or. Style aligned image generation via shared attention. *arXiv preprint arXiv:2312.02133*, 2023.

[8] Kibeom Hong et al. Aespa-net: Aesthetic pattern-aware style transfer networks. In *ICCV*, 2023.

[9] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*, pages 423–439. Springer, 2022.

[10] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[11] Andrey Voynov, Qinghao Chu, Daniel Cohen-Or, and Kfir Aberman. $p+$: Extended textual conditioning in text-to-image generation. *arXiv preprint arXiv:2303.09522*, 2023.

[12] Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 23174–23184, October 2023.

[13] Yuxin Zhang, Weiming Dong, Fan Tang, Nisha Huang, Haibin Huang, Chongyang Ma, Tong-Yee Lee, Oliver Deussen, and Changsheng Xu. Prospect: Expanded conditioning for the personalization of attribute-aware image generation. *arXiv preprint arXiv:2305.16225*, 2023.