Composer: Creative and Controllable Image Synthesis with Composable Conditions

Lianghua Huang [1] Di Chen [1] Yu Liu [1] Yujun Shen [2] Deli Zhao [1] Jingren Zhou [1]

Abstract

Recent large-scale generative models learned on big data are capable of synthesizing incredible images yet suffer from limited controllability. This work offers a new generation paradigm that allows flexible control of the output image, such as spatial layout and palette, while maintaining the synthesis quality and model creativity. With *compositionality* as the core idea, we first decompose an image into representative factors, and then train a diffusion model with all these factors as the conditions to recompose the input. At the inference stage, the rich intermediate representations work as composable elements, leading to a huge design space (*i.e.*, exponentially proportional to the number of decomposed factors) for customizable content creation. It is noteworthy that our approach, which we call **Composer**, supports various levels of conditions, such as text description as the *global* information, depth map and sketch as the *local* guidance, color histogram for *low-level* details, *etc.* Besides improving controllability, we confirm that Composer serves as a general framework and facilitates a wide range of classical generative tasks without retraining. Code and models will be made available.

1. Introduction

> *"The infinite use of finite means."*
>
> – Noam Chomsky (Chomsky, 1965)

Generative image models conditioned on text can now produce photorealistic and diverse images (Ramesh et al., 2022; Saharia et al., 2022; Rombach et al., 2021; Yu et al., 2022; Chang et al., 2023). To further achieve customized generation, many recent works extend the text-to-image models by introducing conditions such as segmentation maps (Rombach et al., 2021; Wang et al., 2022b; Couairon et al., 2022), scene graphs (Yang et al., 2022), sketches (Voynov et al., 2022), depthmaps (stability.ai, 2022), and inpainting masks (Xie et al., 2022; Wang et al., 2022a), or by finetuning the pretrained models on a few subject-specific data (Gal et al., 2022; Mokady et al., 2022; Ruiz et al., 2022). Nevertheless, these models still provide only a limited degree of controllability for designers when it comes to using them for practical applications. For example, generative models often struggle to accurately produce images with specifications for semantics, shape, style, and color all at once, which is common in real-world design projects.

We argue that the key to controllable image generation relies not only on conditioning, but even more significantly on **compositionality** (Lake et al., 2017). The latter can exponentially expand the control space by introducing an enormous number of potential combinations (*e.g.,* a hundred images with eight representations each yield about $\mathbf{100^8}$ combinations). Similar concepts are explored in the fields of language and scene understanding (Keysers et al., 2019; Johnson et al., 2016), where the compositionality is termed *compositional generalization*, the skill of recognizing or generating a potentially infinite number of novel combinations from a limited number of known components.

In this work, we build upon the above idea and present Composer, a realization of *compositional generative models*. By *compositional generative models*, we refer to generative models that are capable of seamlessly recombining visual components to produce new images (Figure 1). Specifically, we implement Composer as a multi-conditional diffusion model with a UNet backbone (Nichol et al., 2021). At every training iteration of Composer, there are two phases: in the decomposition phase, we break down images in a batch into individual representations using computer vision algorithms or pretrained models; whereas in the composition phase, we optimize Composer so that it can reconstruct these images

[1]Alibaba Group [2]Ant Group. Correspondence to: Lianghua Huang, Di Chen, Yu Liu <xuangen.hlh, guangpan.cd, ly103369@alibaba-inc.com>, Yujun Shen, Deli Zhao <shenyujun0302, zhaodeli@gmail.com>, Jingren Zhou <jingren.zhou@alibaba-inc.com>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

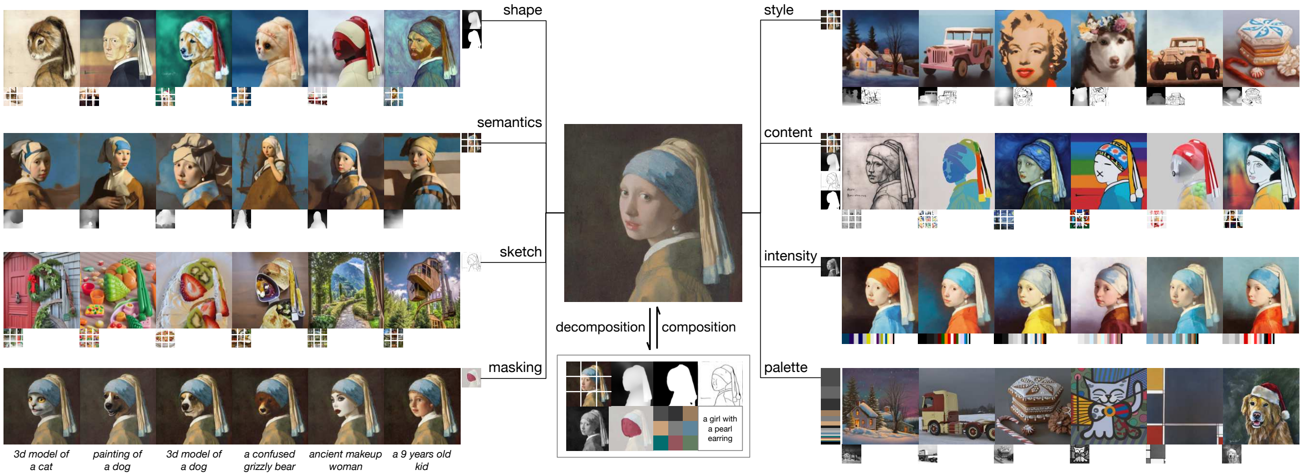


Figure 1. **Concept of compositional image synthesis**, which first decomposes an image to a set of basic components and then recomposes a new one with high creativity and controllability. To this end, the components *in various formats* serve as conditions in the generation process and allow *flexible customization* at the inference stage. Best viewed in large size.

from their representation subsets. Despite being trained with only a reconstruction objective, Composer is capable of decoding novel images from unseen combinations of representations that may come from different sources and potentially incompatible with one another.

While conceptually simple and easy to implement, Composer is surprisingly powerful, enabling encouraging performance on both traditional and previously unexplored image generation and manipulation tasks, including but not limited to: *text-to-image generation, multi-modal conditional image generation, style transfer, pose transfer, image translation, virtual try-on, interpolation and image variation from various directions, image reconfiguration by modifying sketches, depth or segmentation maps, colorization based on optional palettes*, and more. Moreover, by introducing an orthogonal representation of *masking*, Composer is able to restrict the editable region to a user-specified area for *all* the above operations, more flexible than the traditional inpainting operation, while also preventing modification of pixels outside this region. Despite being trained in a multi-task manner, Composer achieves a zero-shot FID of 9.2 in text-to-image synthesis on the COCO dataset (Lin et al., 2014) when using only caption as the condition, indicating its ability to produce high-quality results.

2. Method

Our framework comprises the *decomposition* phase, where an image is divided into a set of independent components; and the *composition* phase, where the components are reassembled utilizing a conditional diffusion model. We first give a brief introduction to diffusion models and the guidance directions enabled by Composer. Subsequently, we explain the implementation of image decomposition and composition in details.

2.1. Diffusion Models

Diffusion models (Ho et al., 2020; Dhariwal & Nichol, 2021; Song & Ermon, 2020; Song et al., 2020b; Nichol et al., 2021) are a type of generative models that produce data from Gaussian noise via an iterative denoising process. Typically, a simple mean-squared error is used as the denoising objective:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{\mathbf{x}_0,\mathbf{c},\boldsymbol{\epsilon},t}(\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(a_t\mathbf{x}_0 + \sigma_t\boldsymbol{\epsilon}, \mathbf{c})\|_2^2), \tag{1}$$

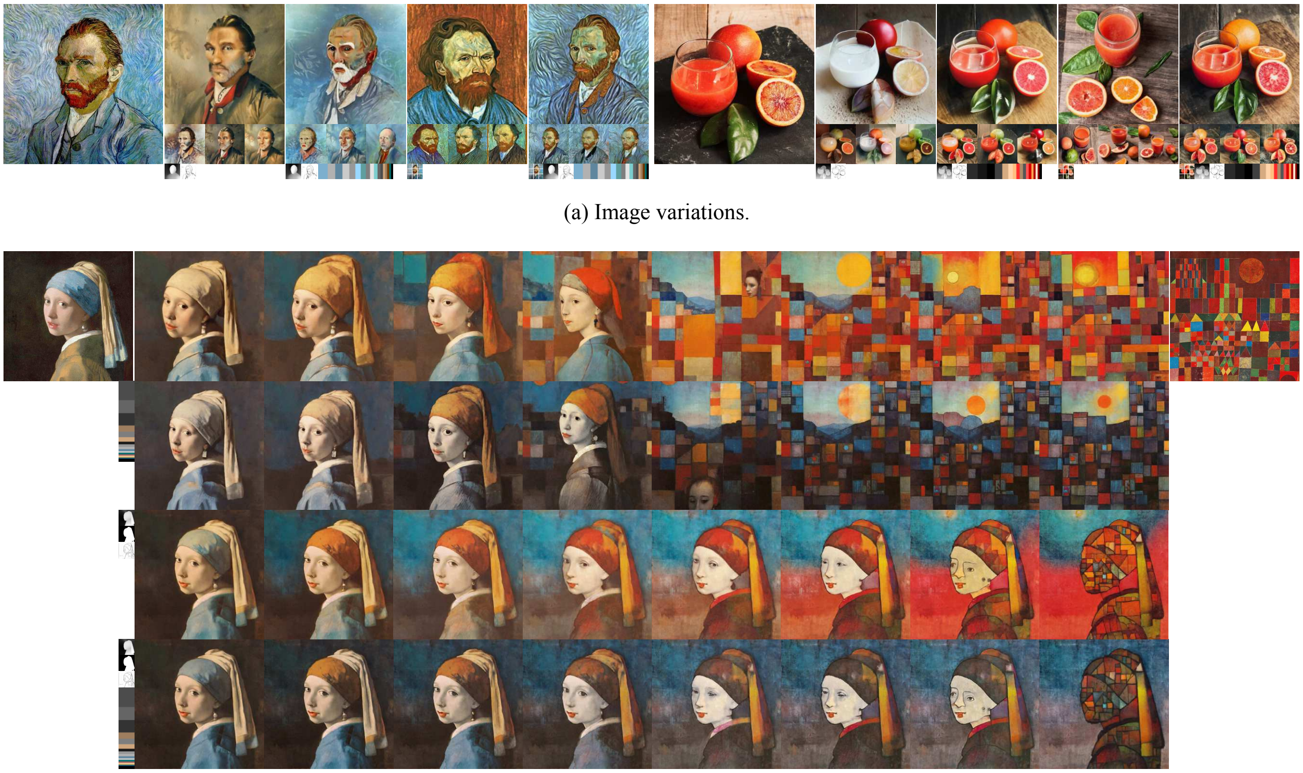
where $\mathbf{x}_0$ are training data with optional conditions $\mathbf{c}$, $t \sim \mathcal{U}(0,1)$, $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ is the additive Gaussian noise, a_t, σ_t are scalar functions of t, and $\boldsymbol{\epsilon}_\theta$ is a diffusion model with learnable parameters θ. *Classifier-free guidance* is most widely employed in recent works (Nichol et al., 2021; Ramesh et al., 2022; Rombach et al., 2021; Saharia et al., 2022) for conditional data sampling from a diffusion model, where the predicted noise is adjusted via:

$$\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, \mathbf{c}) = \omega\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{c}) + (1-\omega)\boldsymbol{\epsilon}_\theta(\mathbf{x}_t), \tag{2}$$

where $\mathbf{x}_t = a_t\mathbf{x}_0 + \sigma_t\boldsymbol{\epsilon}$, and ω is a guidance weight. Sampling algorithms such as DDIM (Song et al., 2020a) and DPM-Solver (Lu et al., 2022a;b; Bao et al., 2022) are often adopted to speed up the sampling process of diffusion models. DDIM can also be utilized to deterministically reverse a sample $\mathbf{x}_0$ back to its pure noise latent $\mathbf{x}_T$, enabling various image editing operations.

Guidance directions: Composer is a diffusion model accepting multiple conditions, which enables various directions with classifier-free guidance:

$$\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, \mathbf{c}) = \omega\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{c}_2) + (1-\omega)\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{c}_1), \tag{3}$$



(a) Image variations.

(b) Image interpolations.

Figure 2. (a) **Image variation.** For each example, the first column shows the source image, while the subsequent four columns are variations of the source image produced by conditioning Composer on different subsets of its representations. (b) **Image interpolation.** On the first row are the results of interpolating all the components between the source image (first column) and the target image (last column). The remaining rows stand for the results where some components (*i.e.*, listed on the left) of the source image are kept unchanged.

where $\mathbf{c}_1$ and $\mathbf{c}_2$ are two sets of conditions. Different choices of $\mathbf{c}_1$ and $\mathbf{c}_2$ represent different emphasis on conditions. Conditions within $(\mathbf{c}_2 \setminus \mathbf{c}_1)$ are emphasized with a guidance weight of ω, those within $(\mathbf{c}_1 \setminus \mathbf{c}_2)$ are suppressed with a guidance weight of $(1 - \omega)$, and conditions within $\mathbf{c}_1 \cap \mathbf{c}_2$ are given a guidance weight of 1.0.

Bidirectional guidance: By reversing an image $\mathbf{x}_0$ to its latent $\mathbf{x}_T$ using condition $\mathbf{c}_1$, and then sampling from $\mathbf{x}_T$ using another condition $\mathbf{c}_2$, we are able to manipulate the image in a disentangled manner using Composer, where the manipulation direction is defined by the difference between $\mathbf{c}_2$ and $\mathbf{c}_1$. Similar scheme is also used in (Wallace et al., 2022). We use this approach in Section 3.2 and Section 3.3.

2.2. Decomposition

We decompose an image into decoupled representations which capture various aspects of its visual concepts. We use eight representations in this work, with all of them extracted on-the-fly during training.

Caption: We directly use title or description information in image-text training data (*e.g.,* LAION-5B (Schuhmann et al., 2022)) as image captions. It is also handy to leverage pretrained image captioning models when annotations are not available. We represent these captions using their sentence and word embeddings extracted by the pretrained CLIP ViT-L/14@336px (Radford et al., 2021) model.

Semantics and style: We use the image embedding extracted by the pretrained CLIP ViT-L/14@336px (Radford et al., 2021) model to represent the semantics and style of an image, similar to unCLIP (Ramesh et al., 2022).

Color: We represent the color statistics of an image using the smoothed CIELab histogram (Sergeyk, 2016). We quantize the CIELab color space to 11 hue values, 5 saturation values, and 5 light values, and we use a smoothing sigma of 10. We find these settings work well empirically.

Sketch: We apply an edge detection model (Su et al., 2021) followed by a sketch simplification algorithm (Simo-Serra et al., 2017) to extract the sketch of an image. Sketches capture local details of images and have less semantics.

Instances: We apply instance segmentation on an image using the pretrained YOLOv5 (Jocher, 2020) model to extract its instance masks. Instance segmentation masks

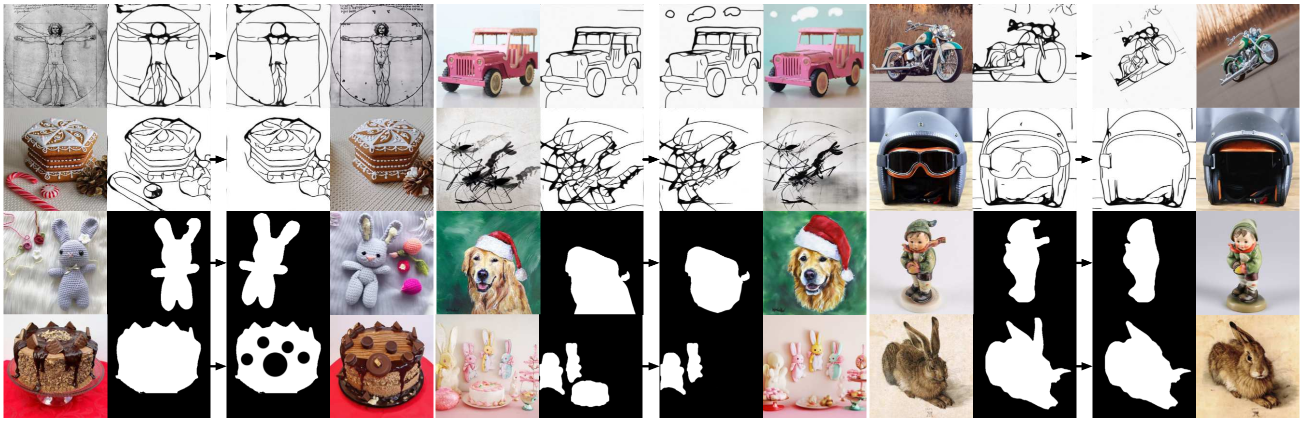


Figure 3. **Image reconfiguration.** Composer supports reconfiguring an image simply by altering its representations, such as sketch and segmentation map.

reflect the category and shape information of visual objects.

Depthmap: We use a pretrained monocular depth estimation model (Ranftl et al., 2022) to extract the depthmap of an image, which roughly captures the image's layout.

Intensity: We introduce raw grayscale images as a representation to force the model to learn a disentangled degree of freedom for manipulating colors. To introduce randomness, we uniformly sample from a set of predefined RGB channel weights to create grayscale images.

Masking: We introduce image masks to enable Composer to restrict image generation or manipulation to an editable region. We use a 4-channel representation, where the first 3 channels correspond to the masked RGB image, while the last channel corresponds to the binary mask.

It is noteworthy that, while the experiments in this paper is conducted with the eight conditions described above, users are free to customize their conditions according to their specific needs.

2.3. Composition

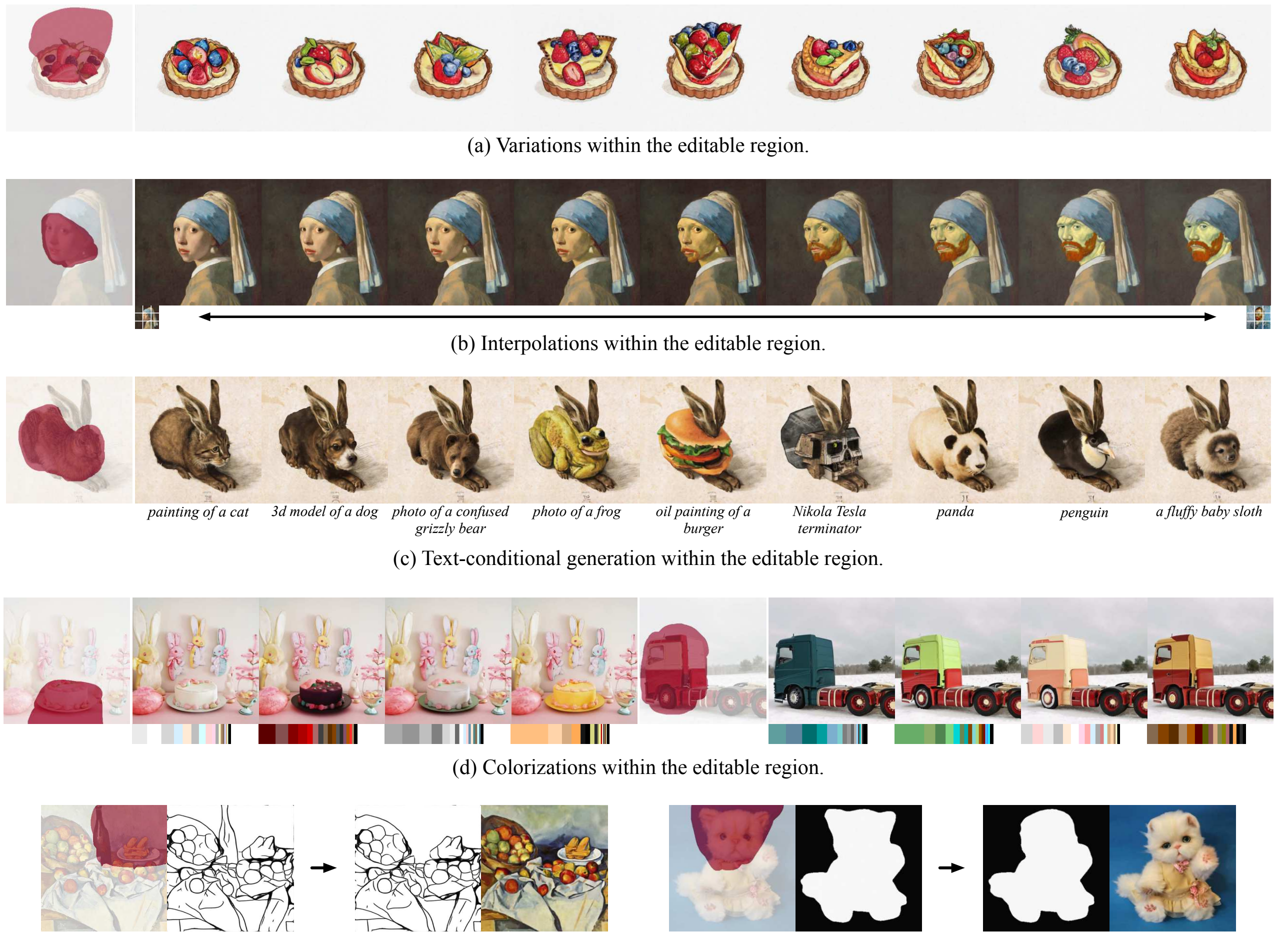
We use diffusion models to *recompose* images from a set of representations. Specifically, we leverage the GLIDE (Nichol et al., 2021) architecture and modify its conditioning modules. We explore two different mechanisms to condition the model on our representations:

Global conditioning: For global representations including CLIP sentence embeddings, image embeddings and color palettes, we project and add them to the timestep embedding. In addition, we project image embeddings and color palettes into eight extra tokens and concatenate them with CLIP word embeddings, which are then used as the context for cross-attention in GLIDE, similar to unCLIP (Ramesh et al., 2022). Since conditions are either additive or can be selectively masked in cross-attention, it is straightforward to either drop conditions during training and inference, or to introduce new global conditions.

Localized conditioning: For localized representations including sketches, segmentation masks, depthmaps, intensity images, and masked images, we project them into uniform-dimensional embeddings with the same spatial size as the noisy latent $\mathbf{x}_t$ using stacked convolutional layers. We then compute the sum of these embeddings and concatenate the result to $\mathbf{x}_t$ before feeding it into the UNet. Since the embeddings are additive, it is easy to accommodate for missing conditions or to incorporate new localized conditions.

Joint training strategy: It is essential to devise a joint training strategy that enables the model to learn to decode images from a variety of combinations of conditions. We experiment with several configurations and identify a simple yet effective configuration, where we use an independent dropout probability of 0.5 for each condition, a probability of 0.1 for dropping all conditions, and a probability of 0.1 for retaining all conditions. We use a special dropout probability of 0.7 for intensity images because they contain the vast majority of information about the images and may underweight other conditions during training.

The base diffusion model produces images of 64×64 resolution. To generate high-resolution images, we train two unconditional diffusion models for upsampling to respectively upscale images from 64×64 to 256×256 and from 256×256 to 1024×1024 resolutions. The architectures of the upsampling models are modified from unCLIP (Ramesh et al., 2022), where we use more channels in low-resolution layers and introduce self-attention blocks to scale up the capacity. We also introduce an optional prior model (Ramesh et al., 2022) that produces image embeddings from captions. We find that the prior model is



(a) Variations within the editable region.

(b) Interpolations within the editable region.

(c) Text-conditional generation within the editable region.

(d) Colorizations within the editable region.

(e) Reconfigurations within the editable region.

Figure 4. **Region-specific image editing.** Through introducing a masked image as an additional condition, Composer manages to direct the manipulation to the region of interest.

capable of improving the diversity of generated images for certain combinations of conditions.

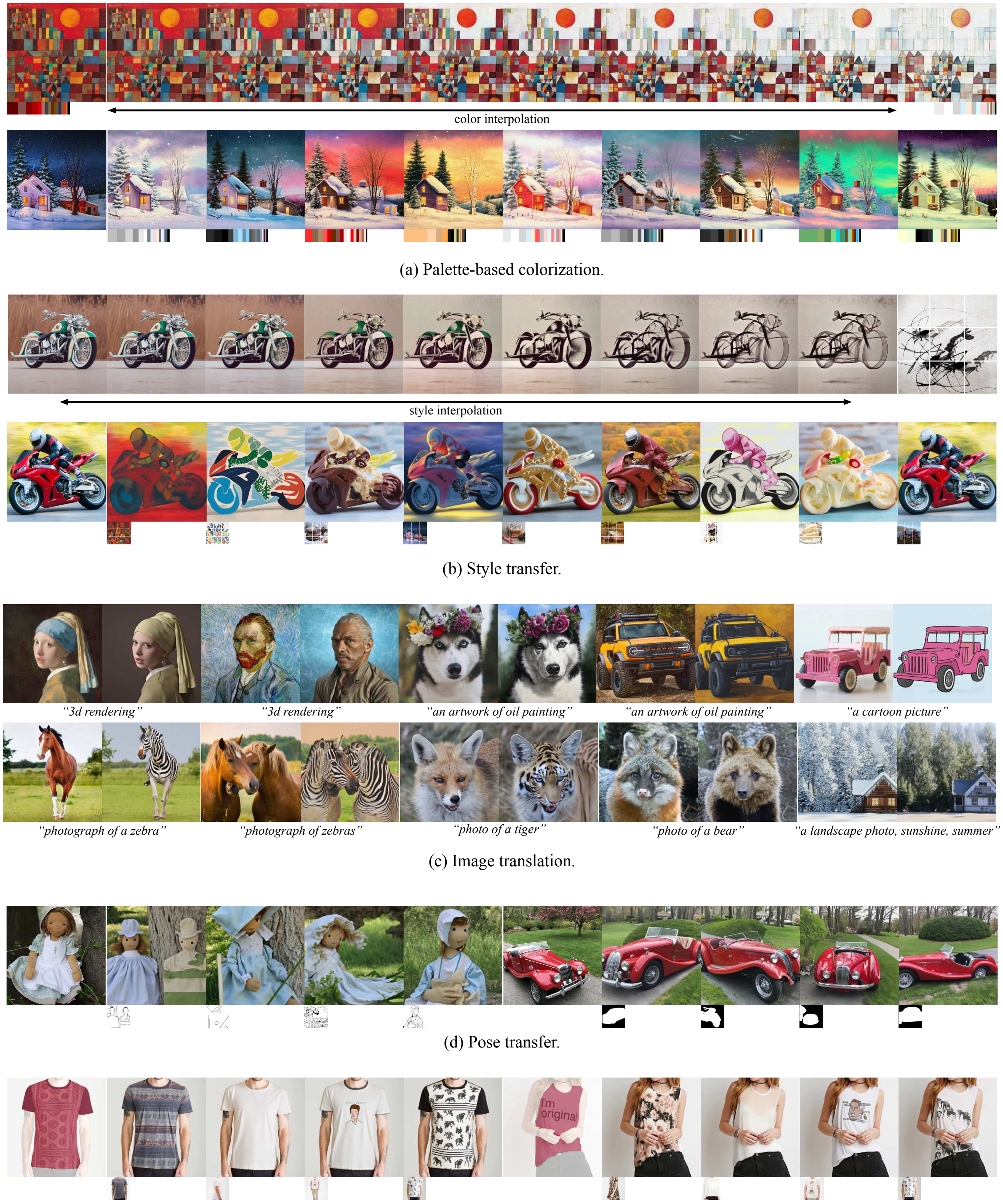
3. Experiments

3.1. Training Details

We train a 2B parameter base model for conditional image generation at 64 × 64 resolution, a 1.1B parameter model for upscaling images to 256 × 256 resolution, and a 300M parameter model for further upscaling images to 1024 × 1024 resolution. Additionally, we trained a 1B parameter prior model for optionally projecting captions to image embeddings. We use batch sizes of 4096, 1024, 512, and 512 for the prior, base, and two upsampling models, respectively. We train on a combination of public datasets, including ImageNet21K (Russakovsky et al., 2014), WebVision (Li et al., 2017), and a filtered version of the LAION dataset (Schuhmann et al., 2022) with around 1B images. We eliminate duplicates, low resolution images, and images potentially contain harmful content from the LAION dataset. For the base model, we pretrain it with 1M steps on the full dataset using only image embeddings as the condition, and then finetune the model on a subset of 60M examples (excluding LAION images with aesthetic scores below 7.0) from the original dataset for 200K steps with all conditions enabled. The prior and upsampling models are trained for 1M steps on the full dataset.

3.2. Image Manipulation

Variations: Using Composer, we can create new images that are similar to a given image but vary in certain aspects by conditioning on a specific subset of its representations. By carefully selecting combinations of different representations, we have a high degree of flexibility to control the scope of image variations (Figure 2a). When more conditions are incorporated, our approach easily yields more accurate reconstructions than unCLIP (Ramesh et al., 2022), which is conditioned solely on image embeddings.



(a) Palette-based colorization.

(b) Style transfer.

(c) Image translation.

(d) Pose transfer.

(e) Virtual try-on.

Figure 5. **Reformulation of traditional image generation tasks** using our Composer. Note that the model is directly applied to all tasks without any retraining, highlighting the potential and flexibility of the proposed compositional generation framework.

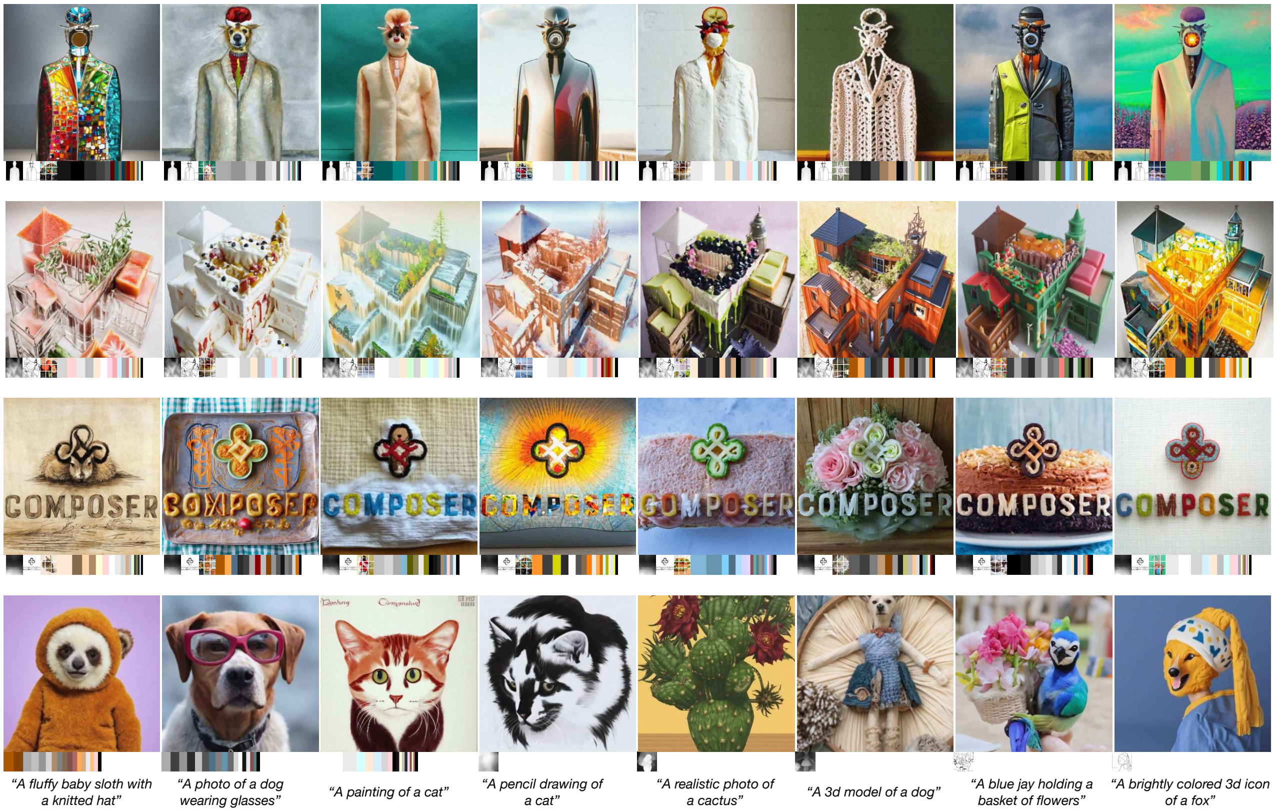

"A fluffy baby sloth with a knitted hat"
"A photo of a dog wearing glasses"
"A painting of a cat"
"A pencil drawing of a cat"
"A realistic photo of a cactus"
"A 3d model of a dog"
"A blue jay holding a basket of flowers"
"A brightly colored 3d icon of a fox"

Figure 6. **Compositional image generation** results produced by Composer. The conditions used to generate each image are presented below the image. Best viewed in large size.

Interpolations: By traversing in the embedding space of global representations between two images, we can blend the two images for variations. Composer further gives us precise control over which elements to interpolate between two images and which to keep unchanged, resulting in a multitude of interpolation directions (Figure 2b).

Reconfigurations: Image reconfiguration (Sun & Wu, 2019) refers to manipulating an image through direct modification of one or more of its representations. Composer offers a variety of options for image reconfiguration (Section 2.1). Specifically, given an image $\mathbf{x}$, we can obtain its latent $\mathbf{x}_T$ by applying DDIM inversion conditioned on a set of its representations $\mathbf{c}_i$; we then apply DDIM sampling starting from $\mathbf{x}_T$ conditioned on a modified set of representations $\mathbf{c}_j$ to obtain a variant of the image $\hat{\mathbf{x}}$. The variant $\hat{\mathbf{x}}$ is expected to differ from $\mathbf{x}$ along the variation direction defined by the difference between $\mathbf{c}_j$ and $\mathbf{c}_i$, but they are otherwise similar. By following this process, we are able to manipulate an image from diverse directions (Figure 3).

Editable region: By conditioning Composer on a set of representations $\mathbf{c}$ along with a masked image $\mathbf{m}$, it is possible to restrict the variations within the area defined by $\mathbf{m}$. Remarkably, editable region is orthogonal to all image generation and manipulation operations, offering Composer substantially greater flexibility of image editing than mere inpainting (Figure 4).

3.3. Reformulation of Traditional Generation Tasks

Many traditional image generation and manipulation tasks can be reformulated using the Composer architecture. Below we describe several examples.

Palette-based colorization: There are two methods to colorize an image $\mathbf{x}$ according to palette $\mathbf{p}$ using Composer: one entails conditioning the sampling process on both the grayscale version of $\mathbf{x}$ and $\mathbf{p}$, while the other involves applying a *reconfiguration* (Section 2.1) on $\mathbf{x}$ in terms of color palette. We find the latter approach yields more reasonable and diverse results and we use it in Figure 5a.

Style transfer: Composer roughly disentangles the content and style representations, which allows us to transfer the style of image $\mathbf{x}_1$ to another image $\mathbf{x}_2$ by simply conditioning on the style representations of $\mathbf{x}_1$ and the content representations of $\mathbf{x}_2$. It is also possible to control the transfer strength by interpolating style representations between the two images. We show examples in Figure 5b.

Image translation: Image translation refers to the task of transforming an image to a variant with content kept unchanged but style converted to match a target domain. We use all available representations of an image to depict its content, with a text description to capture the target domain. We leverage the reconfiguration approach described in Section 2.1 to manipulate images (Figure 5c).

Pose transfer: The CLIP embedding of an image captures its style and semantics, enabling Composer to modify the pose of an object without compromising its identity. We use the object's segmentation map to represent its pose and the image embedding to capture its semantics, then leverage the reconfiguration approach described in Section 2.1 to modify the pose of the object (Figure 5d).

Virtual try-on: Given a garment image $\mathbf{x}_1$ and a body image $\mathbf{x}_2$, we can first mask the clothes in $\mathbf{x}_2$, and then condition the sampling process on the masked image $\mathbf{m}_2$ along with the CLIP image embedding of $\mathbf{x}_1$ to produce a virtual try-on result (Figure 5e). Despite moderate quality, the results demonstrate the possibilities of Composer to cope with difficult problems with one unified framework.

3.4. Compositional Image Generation

By conditioning Composer on a combination of visual components from different sources, it is possible to produce an enormous number of generation results from a limited set of materials. Figure 6 shows some selected examples.

3.5. Text-to-Image Generation

To further assess Composer's image generation quality, we compare its performance with the state-of-the-art text-to-image generation models on the COCO dataset (Lin et al., 2014). We use sampling steps of 100, 50, and 20 for the prior, base, and 64 × 64 to 256 × 256 upsampling models respectively and a guidance scale of 3.0 for the prior and base models. Despite its multi-task training, Composer achieves an competitive FID score of 9.2 and a CLIP score of 0.28 on COCO, comparable to the best-performing models.

4. Related Work

Diffusion models (Ho et al., 2020; Nichol & Dhariwal, 2021; Dhariwal & Nichol, 2021; Rombach et al., 2021; Nichol et al., 2021; Ramesh et al., 2022; stability.ai, 2022; Saharia et al., 2022) are emerging as a successful paradigm for image generation, outperforming GANs (Xu et al., 2017; Zhu et al., 2019; Zhang et al., 2021) and comparable to autoregressive models (Ramesh et al., 2021; Yu et al., 2021; Esser et al., 2020; Yu et al., 2022; Ding et al., 2021) in terms of fidelity and diversity. Our method builds on recent hierarchical diffusion models (Ramesh et al., 2022; Saharia et al., 2022), where one large diffusion model is used to produce small-resolution images, followed by two relatively smaller diffusion models to upscale the image to higher resolutions. However, unlike these text-to-image models, our method supports composable conditions and exhibits better flexibility and controllability.

Many recent works extend pretrained text-to-image diffusion models to achieve multi-modal or customized generation, typically by introducing conditions such as inpainting masks (Xie et al., 2022; Wang et al., 2022a), sketches (Voynov et al., 2022), scene graphs (Yang et al., 2022), keypoints (Li et al., 2023), segmentation maps (Rombach et al., 2021; Wang et al., 2022b; Couairon et al., 2022), a composition of multiple text descriptions (Liu et al., 2022), and depthmaps (stability.ai, 2022), or by finetuning parameters on a few subject-specific data (Gal et al., 2022; Mokady et al., 2022; Ruiz et al., 2022). This work is also related to GAN-based methods that accept a combination of multiple conditions (Huang et al., 2021). Compared to these approaches, Composer merits the compositionality across conditions, enabling a larger control space and greater flexibility in image generation and manipulations.

5. Conclusion and Discussion

Our decomposition-composition paradigm shows that when conditions are treated as composable elements rather than used independently, the control space of generative models can be greatly expanded. This allows for a broader range of traditional generative tasks to be reformulated using our Composer architecture. Moreover, previously unexplored generative capabilities are revealed, which motivates further research into various decomposition algorithms that can achieve increased controllability. In addition, we present multiple ways to utilize Composer for a range of image generation and manipulation tasks based on classifier-free and bidirectional guidance, giving useful references for future research.

Although we find a simple and empirical configuration for joint training of multiple conditions in Section 2.3, the strategy is not perfect, *e.g.,* it may downweight the single-conditional generation performance. For example, without access to global embeddings, sketch or depth based generation usually produces relatively darker images. Another issue is the conflict of the incompatible conditions. For instance, text embeddings are often downweighted in generated results when image and text embeddings with different semantics are jointly used.

Previous studies (Nichol et al., 2021; Ramesh et al., 2022; Saharia et al., 2022) highlight the potential risks associated with image generation models, such as deceptive and harmful content. Composer's improvements in controllability further raise this risk. We intend to investigate thoroughly on how Composer can mitigate the risk of misuse and possibly creating a filtered version before making the work public.

References

Bao, F., Li, C., Zhu, J., and Zhang, B. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. *ArXiv*, abs/2201.06503, 2022.

Chang, H., Zhang, H., Barber, J., Maschinot, A., Lezama, J., Jiang, L., Yang, M., Murphy, K. P., Freeman, W. T., Rubinstein, M., Li, Y., and Krishnan, D. Muse: Text-to-image generation via masked generative transformers. *ArXiv*, abs/2301.00704, 2023.

Chomsky, N. *Aspects of the Theory of Syntax*. The MIT Press, Cambridge, 1965.

Couairon, G., Verbeek, J., Schwenk, H., and Cord, M. Diffedit: Diffusion-based semantic image editing with mask guidance. *ArXiv*, abs/2210.11427, 2022.

Dhariwal, P. and Nichol, A. Diffusion models beat GANs on image synthesis. *ArXiv*, abs/2105.05233, 2021.

Ding, M., Yang, Z., Hong, W., Zheng, W., Zhou, C., Yin, D., Lin, J., Zou, X., Shao, Z., Yang, H., and Tang, J. CogView: Mastering text-to-image generation via Transformers. In *Neural Information Processing Systems*, 2021.

Esser, P., Rombach, R., and Ommer, B. Taming Transformers for high-resolution image synthesis. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12868–12878, 2020.

Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A. H., Chechik, G., and Cohen-Or, D. An image is worth one word: Personalizing text-to-image generation using textual inversion. *ArXiv*, abs/2208.01618, 2022.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *ArXiv*, abs/2006.11239, 2020.

Huang, X., Mallya, A., Wang, T.-C., and Liu, M.-Y. Multimodal conditional image synthesis with product-of-experts GANs. In *European Conference on Computer Vision*, 2021.

Jocher, G. YOLOv5 by Ultralytics, 2020. URL https://github.com/ultralytics/yolov5.

Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C. L., and Girshick, R. B. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1988–1997, 2016.

Keysers, D., Schärli, N., Scales, N., Buisman, H., Furrer, D., Kashubin, S., Momchev, N., Sinopalnikov, D., Stafiniak, L., Tihon, T., Tsarkov, D., Wang, X., van Zee, M., and Bousquet, O. Measuring compositional generalization: A comprehensive method on realistic data. *ArXiv*, abs/1912.09713, 2019.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017.

Li, W., Wang, L., Li, W., Agustsson, E., and Gool, L. V. Webvision database: Visual learning and understanding from web data. *ArXiv*, abs/1708.02862, 2017.

Li, Y., Liu, H., Wu, Q., Mu, F., Yang, J., Gao, J., Li, C., and Lee, Y. J. GLIGEN: Open-set grounded text-to-image generation. *ArXiv*, abs/2301.07093, 2023.

Lin, T.-Y., Maire, M., Belongie, S. J., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, 2014.

Liu, N., Li, S., Du, Y., Torralba, A., and Tenenbaum, J. B. Compositional visual generation with composable diffusion models. *ArXiv*, abs/2206.01714, 2022.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. *ArXiv*, abs/2206.00927, 2022a.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. DPM-Solver++: Fast solver for guided sampling of diffusion probabilistic models. *ArXiv*, abs/2211.01095, 2022b.

Mokady, R., Hertz, A., Aberman, K., Pritch, Y., and Cohen-Or, D. Null-text inversion for editing real images using guided diffusion models. *ArXiv*, abs/2211.09794, 2022.

Nichol, A. and Dhariwal, P. Improved denoising diffusion probabilistic models. *ArXiv*, abs/2102.09672, 2021.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, 2021.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation. *ArXiv*, abs/2102.12092, 2021.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125, 2022.

Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., and Koltun, V. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685, 2021.

Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., and Aberman, K. DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation. *ArXiv*, abs/2208.12242, 2022.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. S., Berg, A. C., and Fei-Fei, L. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2014.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., Salimans, T., Ho, J., Fleet, D. J., and Norouzi, M. Photorealistic text-to-image diffusion models with deep language understanding. *ArXiv*, abs/2205.11487, 2022.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., Schramowski, P., Kundurthy, S., Crowson, K., Schmidt, L., Kaczmarczyk, R., and Jitsev, J. Laion-5b: An open large-scale dataset for training next generation image-text models. *ArXiv*, abs/2210.08402, 2022.

Sergeyk. Rayleigh: Search image collections by multiple color palettes or by image color similarity., 2016. URL https://github.com/sergeyk/rayleigh.

Simo-Serra, E., Iizuka, S., and Ishikawa, H. Mastering sketching: Adversarial augmentation for structured prediction. *arXiv: Computer Vision and Pattern Recognition*, 2017.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *ArXiv*, abs/2010.02502, 2020a.

Song, Y. and Ermon, S. Improved techniques for training score-based generative models. *ArXiv*, abs/2006.09011, 2020.

Song, Y., Sohl-Dickstein, J. N., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *ArXiv*, abs/2011.13456, 2020b.

stability.ai. Stable diffusion 2.0 release., 2022. URL https://stability.ai/blog/stable-diffusion-v2-release.

Su, Z., Liu, W., Yu, Z., Hu, D., Liao, Q., Tian, Q., Pietikäinen, M., and Liu, L. Pixel difference networks for efficient edge detection. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5097–5107, 2021.

Sun, W. and Wu, T. Image synthesis from reconfigurable layout and style. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10530–10539, 2019.

Voynov, A., Aberman, K., and Cohen-Or, D. Sketch-guided text-to-image diffusion models. *ArXiv*, abs/2211.13752, 2022.

Wallace, B., Gokul, A., and Naik, N. EDICT: Exact diffusion inversion via coupled transformations. *ArXiv*, abs/2211.12446, 2022.

Wang, S., Saharia, C., Montgomery, C., Pont-Tuset, J., Noy, S., Pellegrini, S., Onoe, Y., Laszlo, S., Fleet, D. J., Soricut, R., Baldridge, J., Norouzi, M., Anderson, P., and Chan, W. Imagen editor and EditBench: Advancing and evaluating text-guided image inpainting. *ArXiv*, abs/2212.06909, 2022a.

Wang, T., Zhang, T., Zhang, B., Ouyang, H., Chen, D., Chen, Q., and Wen, F. Pretraining is all you need for image-to-image translation. *ArXiv*, abs/2205.12952, 2022b.

Xie, S., Zhang, Z., Lin, Z., Hinz, T., and Zhang, K. SmartBrush: Text and shape guided object inpainting with diffusion model. *ArXiv*, abs/2212.05034, 2022.

Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., and He, X. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1316–1324, 2017.

Yang, L., Huang, Z., Song, Y., Hong, S., Li, G., Zhang, W., Cui, B., Ghanem, B., and Yang, M.-H. Diffusion-based scene graph to image generation with masked contrastive pre-training. *ArXiv*, abs/2211.11138, 2022.

Yu, J., Li, X., Koh, J. Y., Zhang, H., Pang, R., Qin, J., Ku, A., Xu, Y., Baldridge, J., and Wu, Y. Vector-quantized image modeling with improved VQGAN. *ArXiv*, abs/2110.04627, 2021.

Yu, J., Xu, Y., Koh, J. Y., Luong, T., Baid, G., Wang, Z., Vasudevan, V., Ku, A., Yang, Y., Ayan, B. K., Hutchinson, B. C., Han, W., Parekh, Z., Li, X., Zhang, H., Baldridge, J., and Wu, Y. Scaling autoregressive models for content-rich text-to-image generation. *ArXiv*, abs/2206.10789, 2022.

Zhang, H., Koh, J. Y., Baldridge, J., Lee, H., and Yang, Y. Cross-modal contrastive learning for text-to-image generation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 833–842, 2021.

Zhu, M., Pan, P., Chen, W., and Yang, Y. DM-GAN: Dynamic memory generative adversarial networks for text-to-image synthesis. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5795–5803, 2019.

A. Architecture Details

	Prior	64	$64 \rightarrow 256$	$256 \rightarrow 1024$
Diffusion steps	1000	1000	1000	1000
Noise schedule	cosine	cosine	cosine	linear
Sampling steps	100	50	20	10
Sampling variance method	dpm-solver	dpm-solver	dpm-solver	dpm-solver
Model size	1B	2B	1.1B	300M
Channels	-	512	320	192
Depth	-	3	3	2
Channels multiple	-	1,2,3,4	1,2,3,5	1,1,2,2,4,4
Heads channels	-	64	64	-
Attention resolution	-	32,16,8	32,16	-
Dropout	-	0.1	0.1	-
Weight decay	6.0e-2	-	-	-
Batch size	4096	1024	512	512
Iterations	1M	1M	1M	1M
Learning rate	1.1e-4	1.2e-4	1.1e-4	1.0e-4
Adam β_2	0.96	0.999	0.999	0.999
Adam ϵ	1.0e-6	1.0e-8	1.0e-8	1.0e-8
EMA decay	0.9999	0.9999	0.9999	0.9999

Table 1. Hyperparameters for Composer. We use DPM-Solver++ (Lu et al., 2022b) as the sampling algorithm for all diffusion models.

B. Conditioning Modules

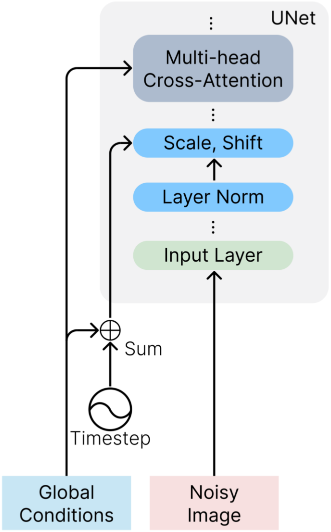


Figure 7. Global conditioning module of Composer. For global conditions such as CLIP sentence embeddings, image embeddings, and color histograms, we project and add them to the timestep embedding. Moreover, we project image embeddings and color palettes into eight extra tokens and concatenate them with CLIP word embeddings, which are then used as the context input for cross-attention layers.

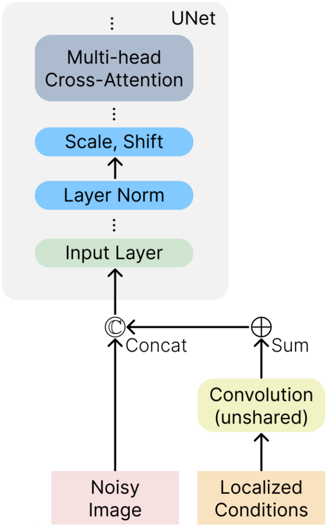


Figure 8. Local conditioning module of Composer. For local conditions such as segmentation maps, depthmaps, sketches, grayscale images, and masked images, we project them into uniform-dimensional embeddings with the same spatial size as the noisy image using stacked convolutional layers. Subsequently, we compute the sum of these embeddings and concatenate the result to the noisy image.

C. Ablation Study of Different Joint Training Strategies

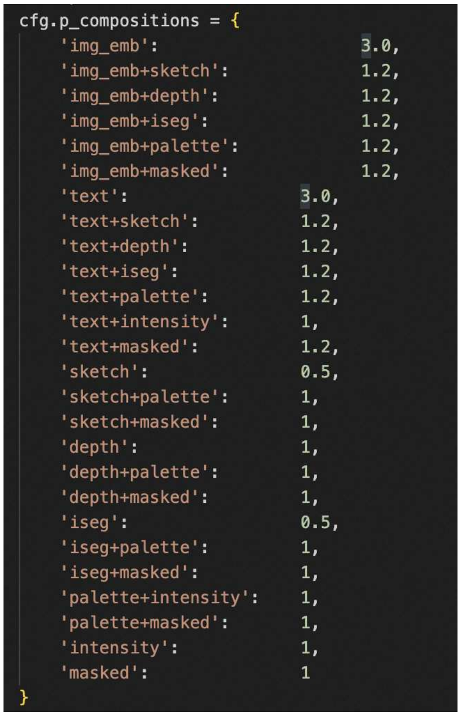
We conduct user studies to evaluate the performance of four pretrained models trained using different settings on five generation tasks. We generate 50 samples per task using the same random seed for each model and seek feedback from a group of participants to identify the best result. Each entry denotes the frequency a participant selected a model as the most favorable result.

The four settings differ in the dropout strategy. The first three settings use an independent dropout for each condition, whereas the final setting involves manually designing sampling probabilities for each individual or paired condition. In the last setting, we hypothesis that the model can learn to produce images from more than two conditions when trained using only single or pair-wise conditions (Figure 9).

The results (Table 2) show that a simple independent dropout of 0.5 for each condition obtains the best overall performance.

Task	**Dropout=0.3**	**Dropout=0.5**	**Dropout=0.7**	**Manually Designed**
Colorization	0.31	0.45	0.11	0.13
Style transfer	0.31	0.41	0.12	0.16
Text-to-image	0.15	0.33	0.16	0.36
Text-and-spatial composition	0.19	0.23	0.36	0.22
Image-embedding-and-spatial composition	0.14	0.34	0.24	0.28
Overall	**0.22**	**0.352**	**0.198**	**0.23**

Table 2. Ablation study of differenct joint training strategies of Composer.



```
cfg.p_compositions = {
    'img_emb':                  3.0,
    'img_emb+sketch':           1.2,
    'img_emb+depth':            1.2,
    'img_emb+iseg':             1.2,
    'img_emb+palette':          1.2,
    'img_emb+masked':           1.2,
    'text':               3.0,
    'text+sketch':        1.2,
    'text+depth':         1.2,
    'text+iseg':          1.2,
    'text+palette':       1.2,
    'text+intensity':     1,
    'text+masked':        1.2,
    'sketch':             0.5,
    'sketch+palette':     1,
    'sketch+masked':      1,
    'depth':              1,
    'depth+palette':      1,
    'depth+masked':       1,
    'iseg':               0.5,
    'iseg+palette':       1,
    'iseg+masked':        1,
    'palette+intensity':  1,
    'palette+masked':     1,
    'intensity':          1,
    'masked':             1
}
```

Figure 9. Manually designed probabilities for different combinations of conditions.

D. Additional Samples

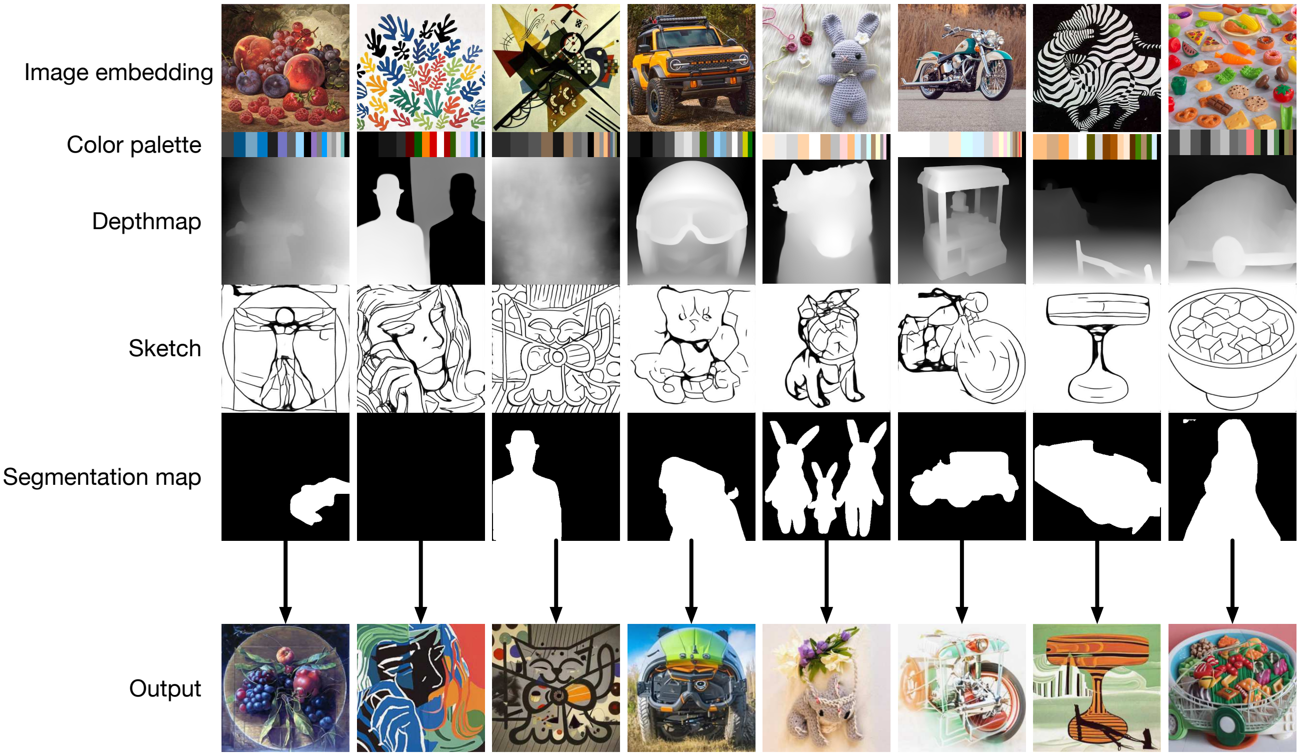


Figure 10. This figure indicates how Composer resolves conflicting conditions by illustrating extreme cases in which the conditions come from disparate sources. One of our observations is that Composer typically gives less weight to conditions with fewer details when conflicts exist, such as segmentation maps in comparison to sketches and depthmaps, and text embeddings versus image embeddings.

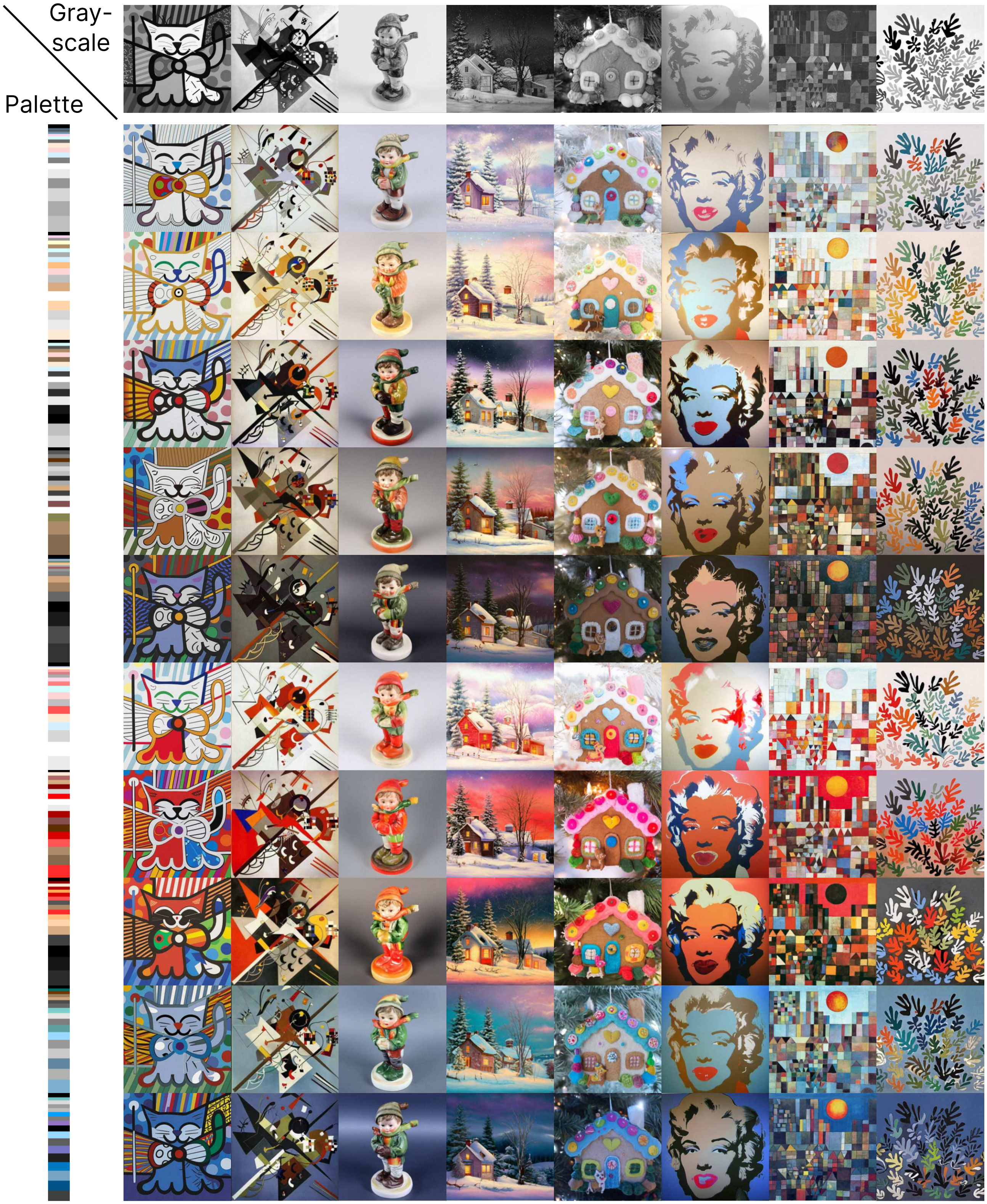


Figure 11. Additional colorization results, visualized at a resolution of 256 × 256 to reduce file size.

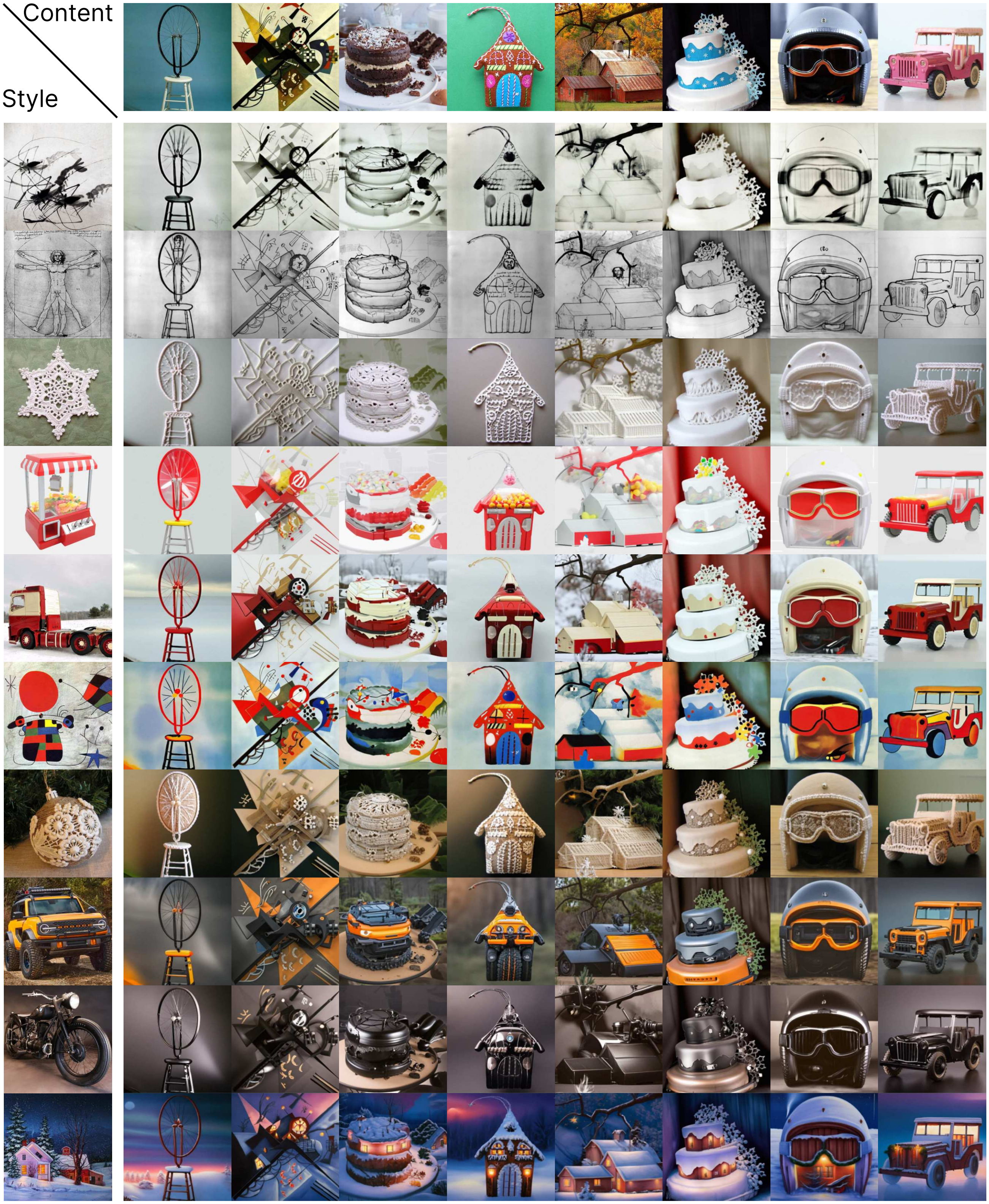


Figure 12. Additional style transfer results, visualized at a resolution of 256 × 256 to reduce file size.

E. Failure Cases of Composer

E.1. Some Conditions are Underweighted When Conflicts Exist

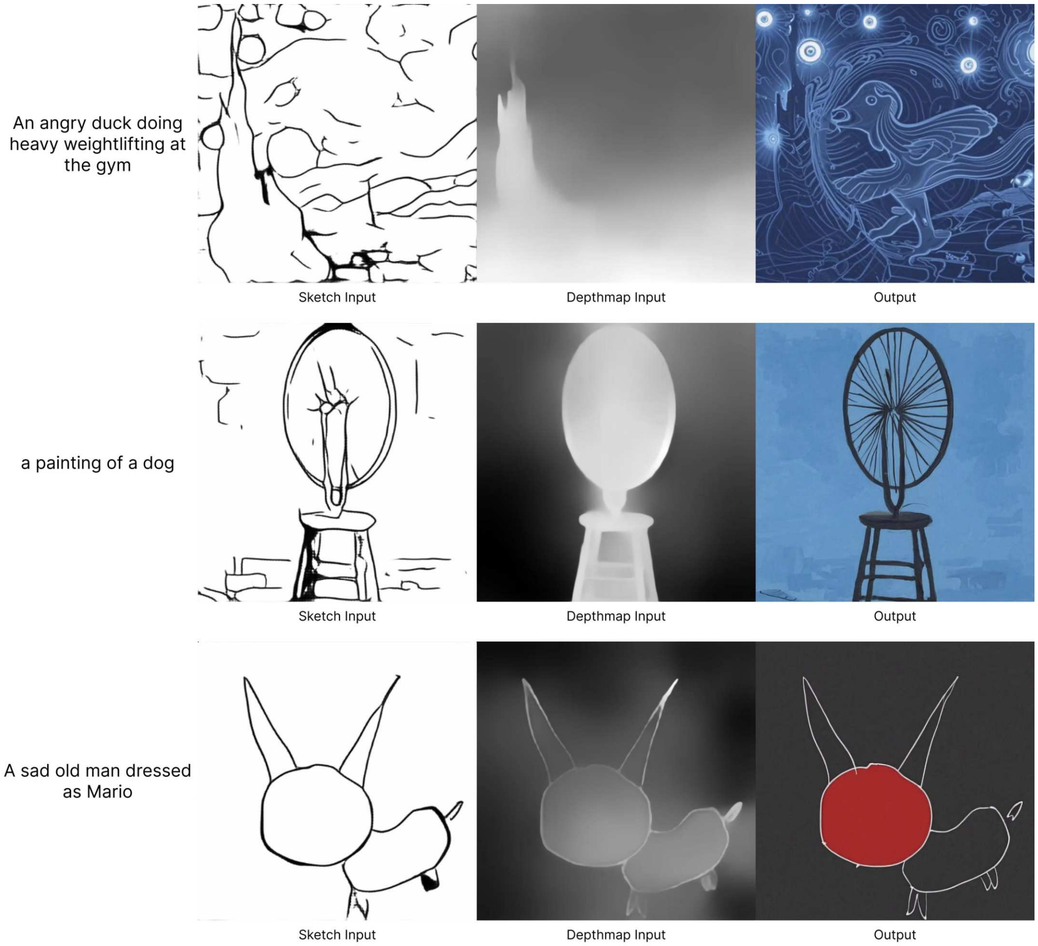


Figure 13. When **text** and **sketch**, **depth** are jointly used as the conditions, **text** conditions are usually underweighted.

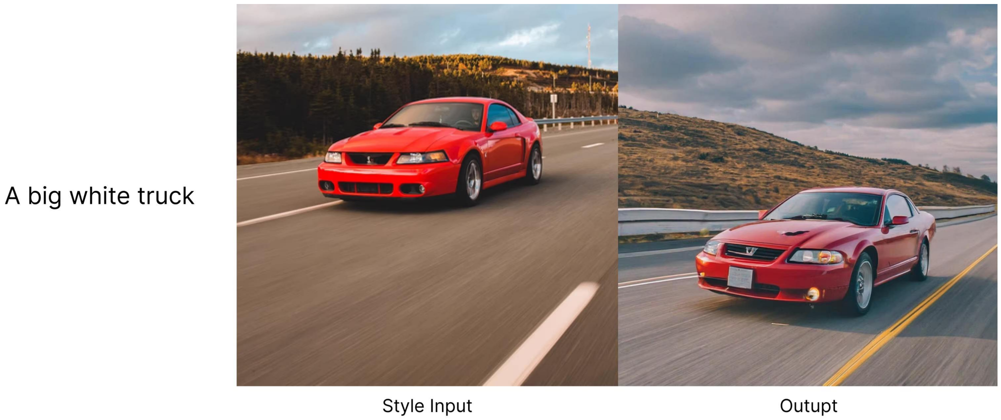


Figure 14. When **text** and **image** embedding are jointly used as the conditions, **text** conditions are usually underweighted.

E.2. Low Quality and Unexpected Results

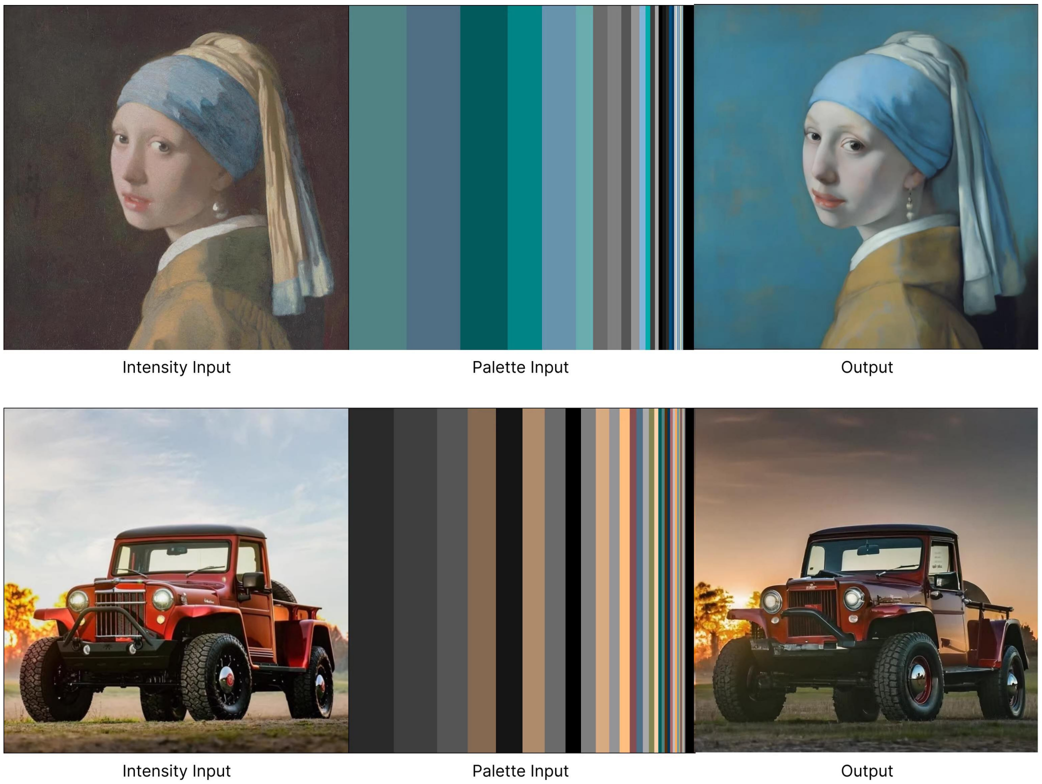


Figure 15. **Colorization** usually changes the background colors or the light conditions rather than the subject colors.

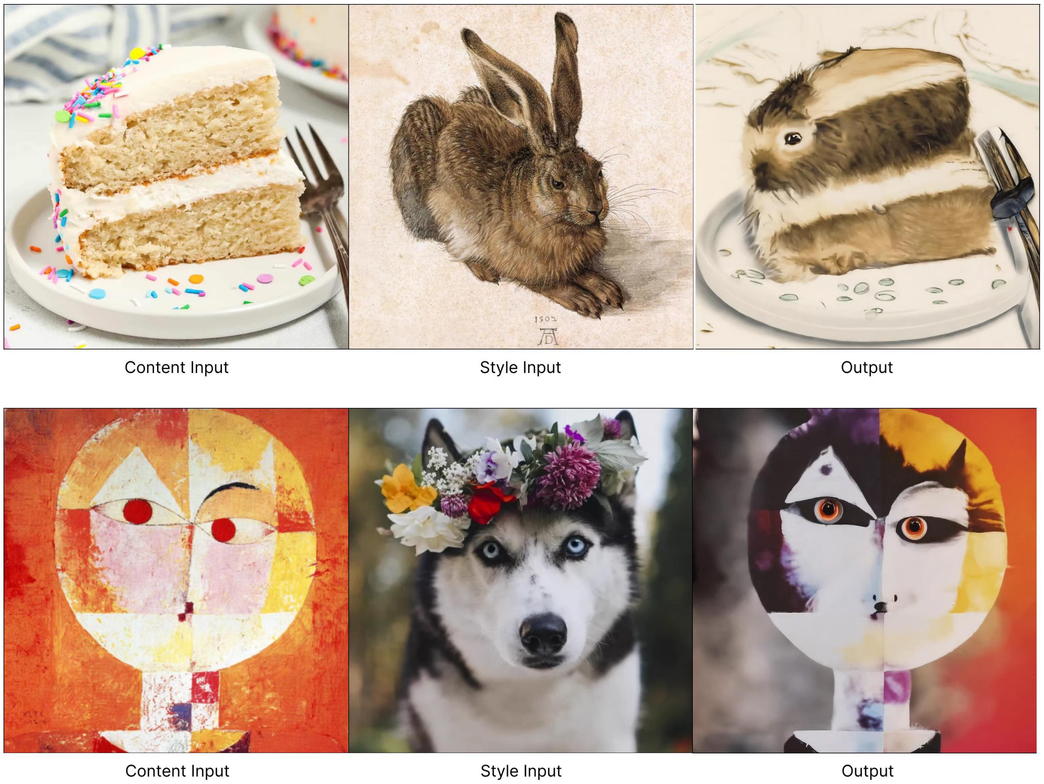


Figure 16. **Style** and **semantics** are usually entangled in **style transfer**.

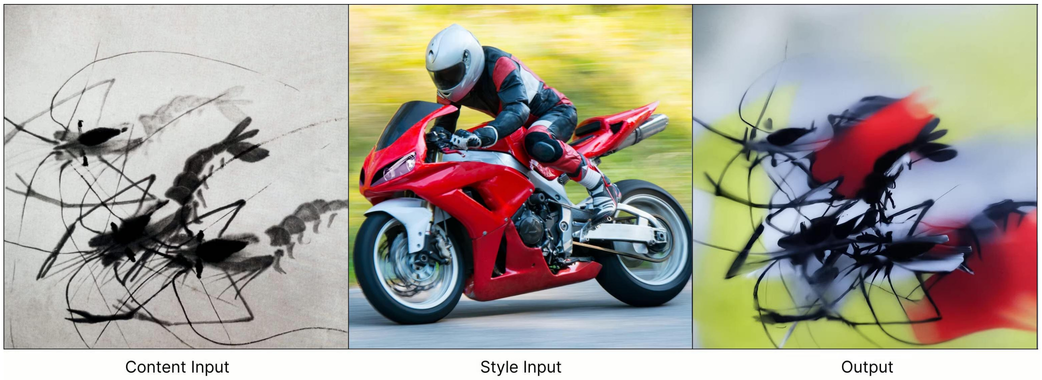


Figure 17. Composer sometimes produces blurry results, especially when conditions are incompatible with each other.

E.3. A Few Tasks may Benefit from Early Stop

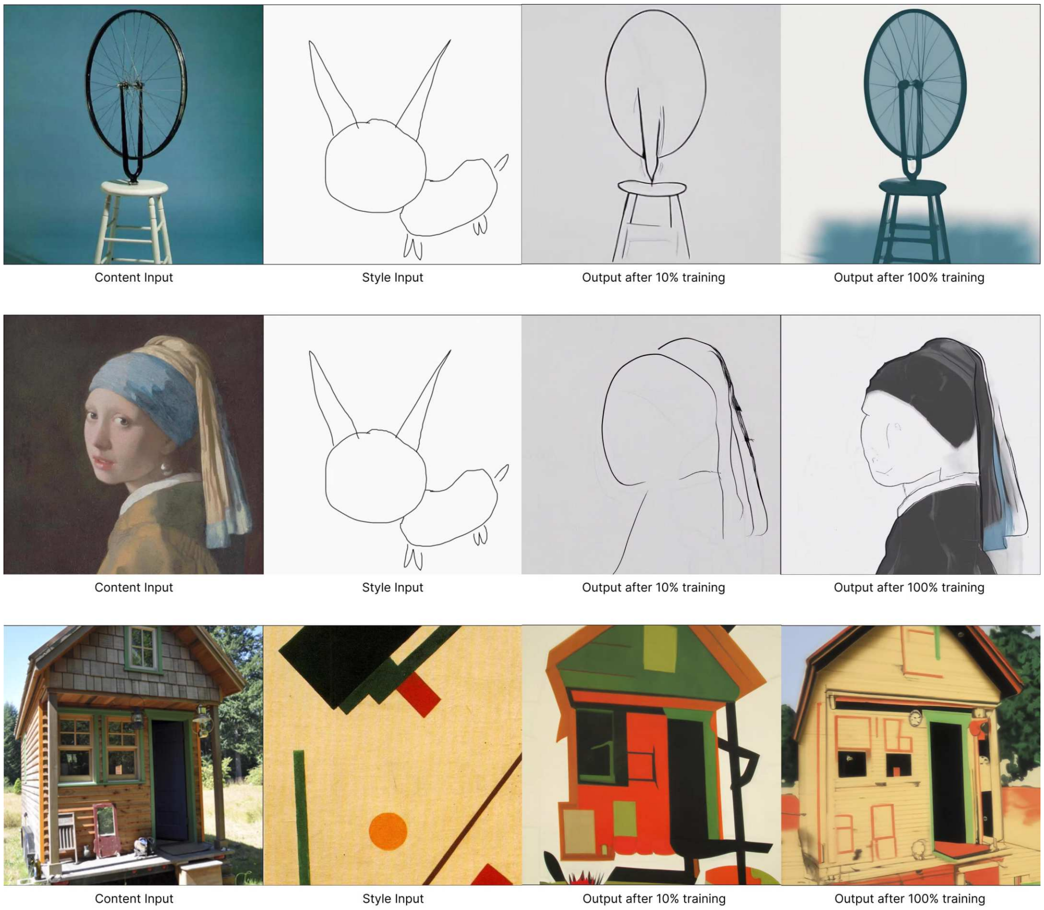

Content Input
Style Input
Output after 10% training
Output after 100% training
Content Input
Style Input
Output after 10% training
Output after 100% training
Content Input
Style Input
Output after 10% training
Output after 100% training

Figure 18. Taking **style transfer** as an example, some style transfer results show better consistency with the style input in the early training stage compared to those outputs from the final training stage.