MCGM-styler: Free-Form styler for Mask Conditional Text-to-Image Generative Model

Rami Skaik, Leonardo Rossi, Tomaso Fontanini, and Andrea Prati

Department of Engineering and Architecture, University of Parma, Italy
{rami.skaik, leonardo.rossi, tomaso.fontanini, andrea.prati}@unipr.it

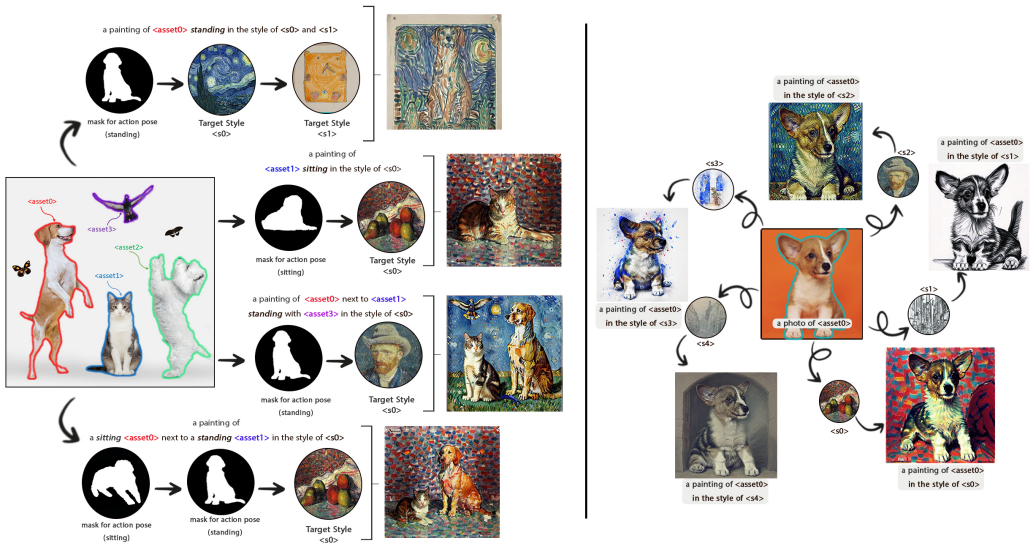

Fig. 1. Overview of the MCGM-Styler capabilities. [On the left] Starting with an image containing multiple subjects, the model is trained sequentially with an action pose mask, a target style, and a textual prompt, enabling users to create scenes with any subject(s) in any pose and style(s). In particular, on the bottom is illustrated the generation of multiple subject in one style, in the middle is illustrated a single subject case, while on the top the use of mixed styles is demonstrated. [On the right] The model generates paintings of a single subject in various styles.

Abstract. Generative models for text-to-image synthesis have made significant advancements in recent years, enabling the creation of highly detailed and stylistically diverse images. In this work, we introduce MCGM-Styler, an extension of our previous MCGM [1] model, which generates images based on masked conditions to specify the action pose of subjects in a source image. Our key contribution is the addition of a new training step that enables the model to also perform style transfer, allowing it to generate images that not only force the pose by mask condition, but also adhere to both single and multiple target artistic styles. Unlike traditional approaches

that require large datasets, MCGM-Styler is trained on a single image, making it highly efficient and adaptable. The model can handle scenes with one or multiple subjects, generating coherent and stylistically consistent outputs so the user can generate any subject(s) in any pose and style or generate an image with a mix of different styles. We evaluate our approach against existing works, particularly the DreamStyler model [2], a state-of-the-art method for style transfer. Our results demonstrate that the MCGM-Styler achieves superior performance in preserving not only the pose of the concept, but also the style fidelity, highlighting its effectiveness in controllable image generation. Code is available at https://github.com/roskaik432/MCGM-Styler.

Keywords: Fine-tuning, Diffusion models, Generative models, Mask Condition, Style transfer

1 Introduction

Generative models play a crucial role in artificial intelligence by synthesizing realistic data from many domains [3], such as images [4,5], text [6,7], and audio [8,9]. These models approximate data distributions and generate samples that closely resemble real-world data. One compelling application of generative models is text-to-image generation, where models synthesize images based on textual descriptions [10]. Modern systems, such as DALL-E 3 [11], utilize large-scale transformer models to successfully bridge text and visual modalities [12]. Over time, researchers have created a variety of strategies and models to generate images from text data, each with its own set of strengths and limitations [13].

Recently, diffusion models achieved impressive results in the field of image generation. They learn to produce data by reversing a gradual noise process. Because of their capability to produce diverse patterns and high-quality, high-resolution samples with a stable training process, these models quickly rose to the top of the image generation field. Furthermore, conditional diffusion models produce realistic and varied outputs by learning to capture complex dependencies between the generated images and the conditioning information through training on large datasets [14]. However, the effectiveness of these models is highly dependent on the volume and diversity of training data, which affects their ability to generalize and generate high-quality outputs [15]. As a result, numerous strategies have recently been proposed to improve diffusion models and make them learn one or more concepts using only a few examples [16]. This allows the learned concepts to be generated in various contexts using textual descriptions. In addition, other models can even generate and edit a particular subject using only a single image [1,17].

Textual embeddings have become a cornerstone in the improvement of diffusion models, with textual inversion emerging as a key technique for personalizing generative output. Textual inversion allows diffusion models to learn and represent new concepts by embedding them as words in the latent space of the model [18], allowing precise control over the generation of images that align with specific descriptions or concepts. This technique can capture unique characteristics from a small set of images, effectively reconstructing and manipulating these concepts during generation.

Beyond concept reconstruction, textual embeddings also facilitate style transfer [19,20] and pose control [1]. A promising development in this space is the task of generating **personalized token embeddings**, which involve creating unique and learnable embeddings for specific users, objects, or styles. These embeddings are trained to capture individual-specific characteristics and can be integrated into generative frameworks for enhanced personalization [21]. In this context, DreamBooth [16], textual inversion [18], Break-a-scene [17] and other works allow fine-tuning of diffusion models with custom tokens to generate content aligned with personal or contextual nuances. Such embeddings extend the utility of generative models in domains such as user-generated content [22], custom art creation [23], and style transfer [24].

Recently, the MCGM model [1] introduced a method to incorporate shape information through a mask encoder within the Break-a-Scene framework [17], which generates scene compositions from a single image that contains multiple concepts by placing them at a specified target pose. In this work, we proposed a better and improved version of the MCGM called *MCGM-styler* that is able to apply both shape and style information when generating the learned concept. This is done by adding an additional training phase to the MCGM. In the first training phase, the model learns to reproduce the concepts and to edit their pose using a mask. In the second and new phase, the model is trained to learn a style embedding that can later be used to condition the generation of the concepts. After these two training phases, during inference, the model is able to generate concepts that vary both their pose and their style, greatly improving the flexibility of the system.

To summarize, the contributions of this work are the following:

- A new architecture which allows to both learn to reproduce multiple subjects in a specific pose from a single image and transfer a desired style to the generated samples.
- Our model is capable of effectively combining multiple conditions – including a text description for detailed scene guidance, a mask to define the pose, and a style reference to customize appearance – providing users with enhanced flexibility, precision, and creative control in generating tailored outputs.
- To boost the ability to learn the style, we incorporated a content description during training to help decouple the content and style from the style image.

2 Related Works

Conditional Generative Models: Conditional generative models enhance generative architectures by incorporating auxiliary variables to enable controlled data generation. Conditional Variational Autoencoders (CVAEs) [25] and Conditional GANs (CGANs) [26] guide the outputs using conditions such as labels or attributes.

For instance, CGANs enable class-conditioned image synthesis, as shown in AC-GAN [27], while Pix2Pix [28] performs image-to-image translation, and AttnGAN [29] generates images from textual descriptions. CVAEs enable tasks like label-conditioned digit generation and diverse dialogue generation. Hybrid models, such as CTRL [30] and StyleGAN [31], further extend control for tasks like text generation and fine-grained face synthesis. Recently, text-to-image conditional generative models have advanced significantly, enabling the creation of highly detailed images from textual descriptions. Models like DALL·E [32] and Stable DIffusion [33] demonstrate precise alignment between text and visuals.

Controllable Generation with Mask-based Conditioning: techniques in text-to-image generative models enable precise spatial and semantic control in image synthesis, advancing the quality and versatility of generated outputs.ControlNet [34] was the first approach based on diffusion models that allowed to add spatial conditioning to large text-to-image diffusion models. Masked-Attention Diffusion Guidance [35] and MaskDiffusion [36] refine spatial alignment through attention maps and conditional masks. Muse [37] combines text prompts and fine-grained masks for localized and semantically guided generation, while DiffEdit [38] facilitates semantic image editing by guiding modifications with masks. Other techniques, like Text-Conditioned Sampling [39] and MaskSketch [40], focus on local refinement and spatial layout control, while Text-Guided Object Inpainting [41] and Adversarial Mask Reconstruction [42] enable targeted image inpainting guided by textual descriptions. Additionally, SemFlow [43] proposed a unified framework able to perform both mask-conditioned generation as well as semantic segmentation. Finally, many mask-based conditional models for face generation have been proposed, such as [44,45,46,47,48]. Nevertheless, all of these approaches are not able to learn a subject starting from a single image and, on the other hand, need to be trained on large datasets. For this reason Break-A-Scene (BAS) [17] was proposed, which decomposes complex scenes into individual components using masks to provide fine-grained control over text-to-image synthesis, offering enhanced flexibility to generate scenes for one or multiple subjects from a one-shot image. Moreover, MCGM (Mask Conditional Text-to-Image Generative Model) [1] was recently proposed to extend the BAS model by injecting mask conditions and embedding spatial constraints in diffusion models to guide the text prompt to generate images for the subjects in specific poses.

Style-based conditioning: The field of style-based conditioning in text-to-image generative models has seen significant improvements, focusing on combining text descriptions with visual style elements to create stylized outputs. Notable works include InstantStyle-Plus [49] and StyleDrop [50], which emphasize efficient style transfer and content preservation, while FineStyle [51] and ArtAdapter [52] introduce fine-grained, controllable style customization and advanced encoding techniques. CSGO [53] and Text-Driven Style Transfer [54] enhance semantic consis-

tency and selective control over style elements. Stylized Text-to-Image Generation focuses on applications in art and design, while InstantStyle [49] , DreamStyler [2], UnZipLoRA [55] present effective disentanglement of style and content for enhanced outputs. Finally, Mamba-ST [56] took advantage of the capabilities of state-space models to perform style transfer. Together, these approaches advance the ability to create visually compelling and semantically aligned images by leveraging style-based conditioning. By taking the step of transferring the style, we are able to improve the MCGM model. In fact, MCGM can only change and specify the pose of the subject using the mask condition; however, our approach enables the model to change the style according to the preference of the user. In order to decouple the content and style from the user style image, we employed a style description technique similar to the one of DreamStyler [2], which employs this technique in a sophisticated manner by combining the Blip-2 technique with the style description. In our case, we simplify this procedure by only using human description of the styles.

Personalized token embeddings: The personalization task focuses on capturing and leveraging concepts from exemplar images as generative conditions, particularly those that are difficult to express through text, to enable controllable generation. Personalized token embeddings seek to modify embeddings to represent specific things, concepts, or styles in a customized way. These embeddings are especially valuable for activities like customized generation, style transfer, and fine-tuning huge models to recognize or incorporate user-defined content. In text-to-image (T2I) generative models, many applications used token embeddings to personalize the results. The textual inversion [18] technique is based on open-ended conditional synthesis models. Rather than training a new model from scratch, it finds new pseudo-words in the embedding space of a text encoder, driving personalized creation in an intuitive way. On the other hand, DreamBooth [16] fine-tunes the entire T2I model for better subject representation. Later methods aim to optimize specific parts of the networks [57,58]. These models use several images of a single visual concept to learn to generate this concept in different contexts, while recent models [59,60] can generate new scenes for concepts using only a single reference image; moreover, BAS [17] can extract multiple concepts from a single image. The MCGM [1] model used the same technique as the BAS, but by adding a new mask condition to specify the POSE of the concept in the new scene. Furthermore, techniques such as LoRA [61] and StyleDrop [50] focus on optimizing low-rank approximations and a small subset of weights, respectively, for style personalization. In MCGM-styler, we used the personalized token embeddings to represent the concepts and the style to customize the generated scene.

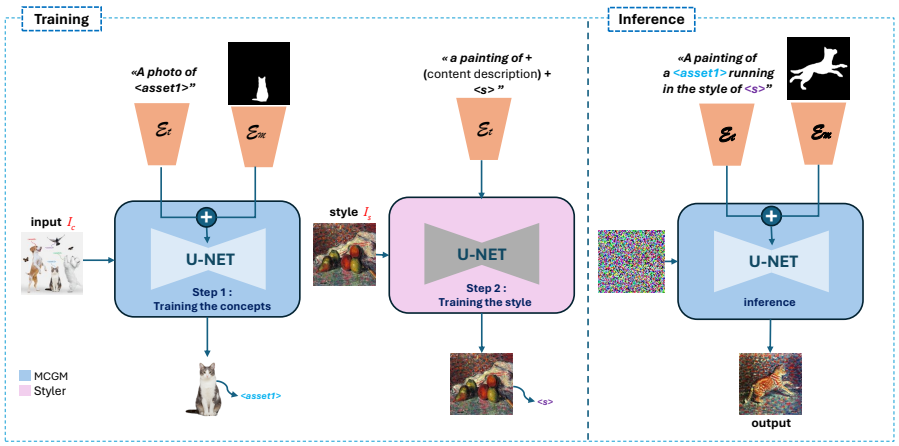

Fig. 2. Pipeline of the MCGM-Styler: The left section illustrates the two-stage training process, with source and style images provided as user inputs. In the first stage, various concepts are learned using textual prompts and mask inputs. In the second stage, the style concept is trained. The token $< asset1 >$ represents the cat in the source image, while $< s >$ represents the style. Note that the $(+)$ symbol indicates concatenation of the text with mask embeddings. The right section depicts the inference process, where a textual prompt description and a target position mask are used to generate the output scene.

3 The Proposed Method

3.1 Model architecture overview

Our model architecture shown in Figure 2 is structured into two main training stages. In the first stage, a text encoder and a mask encoder are used along with a source image to train the diffusion model to learn the pose of the concepts. The second stage focuses on learning stylistic attributes; it employs a diffusion model combined with a text encoder and a target style image. During inference, the model generates a final image starting from random noise, guided by the text and mask encoders. Further details on training and inference procedures are provided later in this section.

Preliminary: our objective is to generate samples for any subject in any style and any pose specified by the user. Our approach is based on our previous work, MCGM [1], which was able to generate a new scene starting from a single image I_c containing one or more objects. The main contribution of the MCGM was the ability to also specify the pose of these objects using a textual prompt and an additional mask condition indicating the target action pose of the object inside the generated scene. MCGM model is built on [17], which customizes a pre-trained text-to-image model by incorporating a distinct embedding to represent the target

concept. Beyond its ability to identify multiple concepts within a single image, MCGM introduces N textual handles, $\{v_i\}_{i=1}^{N}$, where each handle v_i corresponds to the concept represented by the mask M_i. To capture the contextual pose, a mask encoder $\mathcal{E}_m$ is employed to encode each mask in a mask embedding m_i, which is then combined with the corresponding text embedding t_i. Finally, both textual and mask embeddings are fed into the cross-attention layers of the diffusion model.

3.2 Injecting the style of the concepts

Our goal is to improve the MCGM framework by adding a new stage to the training process. This stage involves training the model to integrate a new style token s_*, which can be included in any text prompt to specify the desired style for the generated image.

As shown in Figure 2, our proposed model training process is split into two stages: one for concept training and another for style training. In the first stage, the model takes as input a single image I_c that contains one or more concepts and masks to learn the concept and pose, while in the second stage the input becomes a style image I_s.

Stage one: learn the concepts. In the first stage, the model is trained to identify the concepts from a single input image with the help of the mask encoder $\mathcal{E}_m$ to learn the pose applied. A small dataset of image-text pairs is created in the format "A photo of $[v_x]$ and $[v_y]$..." to extract the textual handles v_i, which represent specific concepts. The background of the input image is then masked using the provided masks. The embeddings that are generated in this stage can then be used with the embeddings of the style in the next stage to generate the scene in the inference stage.

Practically, our model utilizes textual inversion to tailor token embeddings and, additionally, a masked variant of the standard diffusion loss [62], detailed in Equation 1, is applied. The latter focuses on optimizing the handles and model weights by restricting the penalty to the pixels covered by the concept masks:

$$L_{rec1} = \mathbb{E}_{z,s,\epsilon\sim N(0,1),t}\left[\left\|\epsilon \odot M_s - \epsilon_\theta\left(z_t, t, c_s\right) \odot M_s\right\|_2^2\right] \tag{1}$$

where z_t denotes the latent variable with noise added at time step t, s represents the number of concepts, c_s is the set of embeddings obtained from text and masks, M_s corresponds to the set of masks, ϵ indicates the noise, and ϵ_θ refers to the de-noising network.

When relying solely on diffusion loss, the resulting handles struggle to effectively separate the corresponding concepts [17], as there is no mechanism to penalize the association of a single handle with multiple concepts. To address this limitation, it is beneficial to analyze the cross-attention maps between the learned handles and the generated images. For this reason, the MCGM model [1] introduced a cross-attention loss to guide the model not only to reconstruct the pixels of the learned

concepts but also to ensure that each handle focuses exclusively on the image region corresponding to its associated concept. Formally, we define the cross-attention loss as follows:

$$L_{attn} = \mathbb{E}_{z,k,t}\left[\|\mathcal{CA}_\theta(v_i, z_t) - M_{ik}\|_2^2\right] \tag{2}$$

where $\mathcal{CA}_\theta(v_i, z_t)$ is the attention map of the token v_i and the noisy latent z_t obtained from each cross-attention layer, k is the subset of concepts randomly selected during each training steps, and $\mathcal{M}_{ik}$ is the mask for each concept.

Additionally, during the learning of object pose, the pose mask embeddings are combined with textual embeddings for each subject. This integration may introduce ambiguity into the diffusion model, making it difficult to reconstruct each subject individually and possibly resulting in mixed characteristics or uncontrollable scenes. To address this issue, we include a new mask cross-attention loss tailored for mask tokens m_i. Formally, the total cross-attention loss is defined as follows:

$$L_{Mattn} = L_{attn} + \lambda_m \mathbb{E}_{z,k,t}\left[\|\mathcal{CA}_\theta(m_i, z_t) - M_{ik}\|_2^2\right] \tag{3}$$

Here, λ_m represents the weight assigned to the mask cross-attention loss and $\mathcal{CA}_\theta(m_i, z_t)$ denotes the cross-attention map between the mask token m_i and the noisy latent z_t, and $\mathcal{M}_{ik}$ is the mask of the concept k .

As a result, in the first stage of training, the total optimization loss of the first training stage is

$$L_{T1} = L_{rec1} + \lambda_{t1} L_{Mattn} \tag{4}$$

Stage two: learn the style. In the second stage, the MCGM-styler model is finetuned to learn the desired style starting from the model trained in the first stage using the input style image I_s. For this, a collection of image-text pairs is created in the format "A painting of s_* " to learn the textual handler s_* that represents the target style.

Practically, during this training stage, the model is trained to capture the user style. In this case, textual inversion is used to tailor style token embeddings and the standard diffusion loss [62], detailed in Equation 5, is applied:

$$L_{T2} = \mathbb{E}_{z,\epsilon\sim N(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, s_*)\|_2^2\right] \tag{5}$$

Here z_t represents the latent variable with noise added at the time step t, and s_* is the set of embeddings obtained from the style, ϵ indicates the noise, and ϵ_θ refers to the de-noising network.

As illustrated in Figure 2, upon completion of these two stages, the user can create any desired scene featuring specific objects by providing a text prompt, the target pose mask, and the style learned during training.

3.3 Decoupling content and style

The MCGM-styler model still faces a fundamental challenge when training with a style reference: the style and content of the image may become entangled during the training phase. This problem arises mainly when the model attempts to encapsulate all features of the image in s_*, without focusing on only the style aspect. In addition, the absence of contextual information in the training prompt causes the model to ignore the intended context during inference. As shown in Figure 9, elements from the style image, such as the apples and the blanket, are carried over to the output along with the subject, although the intention of the user was to transfer only the style.

DreamStyler [2] resolved the style-context decoupling by adding contextual descriptions to the training prompt. BLIP-2 [63] was used to caption non-style attributes in automatic prompt generation and also included human feedback to improve disentanglement capability in certain style images. In our work, we used a single strategy for decoupling, which is human feedback. Additionally, DreamStyler used textual inversion training with multiple iterations (6 iterations for optimal results), while we only used one.

Practically, we inject a contextual description C into the training prompt to allow the model to better disentangle the style from the context. This is done by designing training prompts that include contextual details about the style image. Let $P = [p_0, s_*]$ represent the standard prompt used in the second stage of the training process, where p_0 is the initial text (e.g., "a painting of") and s_* denotes the style token set described earlier. In our proposed approach, a contextual descriptor p_c (e.g., "pencil, pears, and apples on a cloth, in the style of") is introduced in the middle of the prompt, resulting in $C = [p_0, p_c, s_*]$. **The contextual descriptor is generated by annotating all non-style attributes of the style image, including objects, composition, and background. Finally, an example of a training prompt could be: "A painting of a pencil, pears, and apples on a cloth, in the style of s_*". This textual description of the style facilitates the decoupling of content and composition from the actual style in the image.

This description consists of a concise list of non-style attributes (5–15 words) describing the objects, composition, and background in the style image. Short but specific descriptions (e.g., 'pencil, pears, and apples on a cloth') are sufficient for effective decoupling. Preparing the description requires less than one minute from a non-expert user. An example training prompt is: 'A painting of a pencil, pears, and apples on a cloth, in the style of s*.' This lightweight annotation process offers a practical balance between user effort and model performance.

After training the model using the two-stage approach and including the decoupling technique during the style training stage, our model is able to generate and reconstruct concepts based on the specified pose and style described in the textual prompt, as shown in Figure 2 on the right side. The target pose mask is

injected into the model to specify the pose of the generated subjects, while the text embeddings is used to specify the style.

4 Experiments and Evaluation

4.1 Training details

As detailed in Section 3, the MCGM-Styler training process is composed of two distinct stages. Both stages follow the same setup, which involves a two-phase training approach. Initially, text embeddings are optimized using a high learning rate of 5×10^{-4}. In the subsequent phase, the UNet and text encoder weights are fine-tuned with a lower learning rate of 2×10^{-6}. The training process maintains consistent settings throughout both phases, using the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.99$, and a weight decay of 1×10^{-8}. Each phase of the optimization process consists of 400 steps for a total of 800 steps for each stage.

The learning rates (5×10^{-4} for embedding optimization, 2×10^{-6} for fine-tuning) were chosen after a grid search over $\{1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-}3\}$ for the first phase and $\{1 \times 10^{-6}, 2 \times 10^{-6}, 5 \times 10^{-6}\}$ for the second phase. The selected rates provided the best trade-off between convergence speed and preservation of pretrained weights. For timesteps, we evaluated {400, 600, 800, 1000} and found 800 to yield optimal style fidelity without overfitting. This consistency across both training stages also helped maintain style-concept balance.

Computationally, the model requires $\tilde{8}$GB of GPU memory for inference at 512×512 resolution, and the total training process for concepts and style takes $\tilde{3}2$ minutes on a single NVIDIA A100 GPU.

4.2 Experimental results

Single Concept Generation Across Diverse Styles: The style-conditioned generation task has the objective of extracting style information from the provided samples to condition the generation. Fig. 3 presents several images generated by extracting one concept from a single image and applying five different styles to it, showing the effectiveness of our method to handle different styles.

Single/Multiple Concepts with Guided Pose and Style: This section illustrates the main contribution of our model. As shown in Fig. 4, the MCGM-Styler model allows one to generate images of concepts utilizing both pose and style conditions. The figure demonstrates examples of generated images for one concept with the same textual prompt in two different cases. Firstly, only the style condition is used without pose; in this case, we can see the object (*e.g.*, a cat) playing football with a generic pose. Secondly, both style and pose conditions are employed: we can recognize in the figure that changing the shape of the pose mask each time can change the pose of the subject accordingly.

Fig. 3. One Concept, One Style: Different images are generated from a single input image, representing two distinct concepts. The generation process follows the prompt "a painting of $< asset0 >$ / $< asset1 >$ in the style of $< s0 >$, where $< asset0 >$ refers to the left dog, $< asset1 >$ represents the cat, and $< s0 >$ denotes the target style. The figure highlights the effectiveness of incorporating style conditioning in generating images for both concepts.

Additionally, our model is not limited to generating scenes for a single concept; it can also generate scenes for multiple concepts using one or more masks to guide the pose of the object actions described in the text prompts while still applying the desired style. As illustrated in Fig. 5, we present three different scenarios: one using a single pose mask for one concept (*e.g.*, the sitting pose of the cat in (A)), another using a single pose mask for multiple concepts (*e.g.*, the sitting pose of the cat and the dog in (B) and (C)), and finally, a case using multiple pose masks for multiple concept actions (*e.g.*, one mask for the sitting pose of the dog and another for the sleeping pose of the cat in (D)).

Single concept in mixed styles: Furthermore, our model can learn multiple styles and, as a consequence, generate a single image consisting of a mixture of styles. For example, if a user wants to create an image of a concept incorporating their preferred mixed styles, they can do so by training the model with two or more styles. The trained model can then be used to generate the desired scene using either a single style or a combination of multiple styles.

As shown in Fig. 6, after training the model with two different styles, concepts can be painted using one of these styles or with a mixture of both, depending on the token style holders used in the text prompt during the inference phase (*e.g.*, <asset0>, <asset1> , etc.). Additionally, Fig. 7 presents examples in which different pairs of styles were trained and applied to generate the same text prompt for a single concept. The results illustrate how each pair of styles blends uniquely in the generated images.

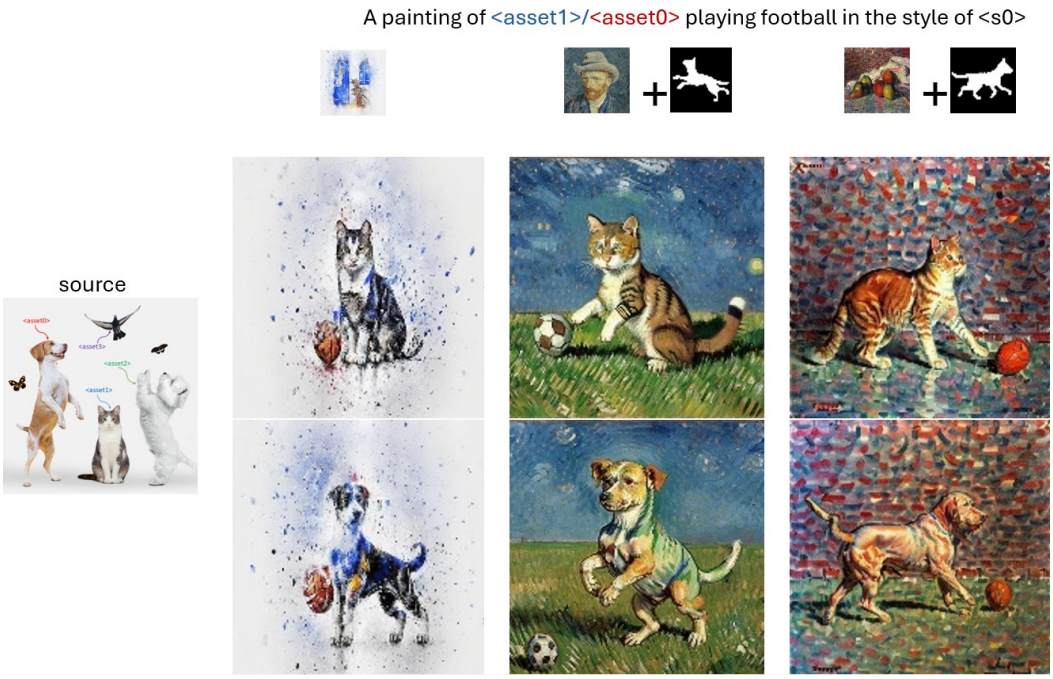

Fig. 4. One concept with varying masks and styles: Generated outputs are based on the same textual prompt, but guided by different styles and mask shapes. The first column uses only style guidance, while the second and third columns incorporate both style and mask guidance. The first row corresponds to $< asset1 >$, representing the cat concept, and the second row corresponds to $< asset0 >$, representing the dog concept.

Comparisons with state of the art: To evaluate the capability of our model to transfer a style to a particular object, we performed experiments to test our model's style transfer performance and compared it in Fig. 8 with the Dreamstyler model [2] in different scenarios. Firstly, in (B), Dreamstyler was trained using MCGM [1] to learn the concept to which the style needs to be applied. In this case, the results were superior to the original DreamStyler model, which utilized DreamBooth as a pre-trained model and required multiple training stages for optimal results. This configuration is shown in column (C). Unlike DreamStyler, which relies on multistage training for style transfer, our model requires only a single stage and still produces better results. To ensure a fair comparison, we also evaluated our single-stage model against DreamStyler trained in a single stage. The results presented in column (A) show that our model outperforms DreamStyler in all scenarios.

4.3 Ablation study

Decoupling style and content: Fig. 9 illustrates the different behaviors of the proposed system when the contextual description of the style is used or not during

Fig. 5. Multi-Concepts-Multi-Masks: (A) illustrates the case of using a single mask to guide only one of the concepts pose, specifically guiding the sitting pose for the cat. (B) demonstrates the case of using a single mask to guide actions for two concepts. In this example, the mask guides the sitting position for both the cat and the dog. (C) showcases the case of generating three concepts in one scene using one mask to guide their pose. Finally, (D) showcases the case of using multiple masks to guide multiple actions for multiple concepts, with one mask guiding a sitting dog and the other guiding a sleeping cat.

the training process. More in detail, our system integrates a new human description prompt in the training process with the objective of decoupling style and content from the style image. This description consists of a concise list of non-style attributes (5–15 words) describing the objects, composition, and background in the style image. Short but specific descriptions (e.g., 'pencil, pears, and apples on a cloth') are sufficient for effective decoupling.

Preparing the description requires less than one minute from a non-expert user. An example training prompt is 'A painting of a pencil, pears, and apples on a cloth, in the style of s*.' This lightweight annotation process offers a practical balance between user effort and model performance. From the figure it is evident how, without this description, the model struggles to generate the concepts and instead copies elements that are present in the style image.

Contribution of Mask Encoder: We have also tested the effect of the mask encoder. As shown in the Fig. 10, removing the mask encoder leads to a reduction in pose accuracy. For example, when using the prompt "a painting of <asset1> playing football in the style of <s0>", the generated object (cat) fails to align its action (playing football) with the target mask pose, demonstrating the importance of the mask encoder.

Different time steps in the training process: Additionally, during the style training phase, we trained the model using different number of timesteps. As shown in Fig. 11, before reaching 800 timesteps, the model fails to accurately apply the style, resulting in generated images that do not fully align with the reference style.

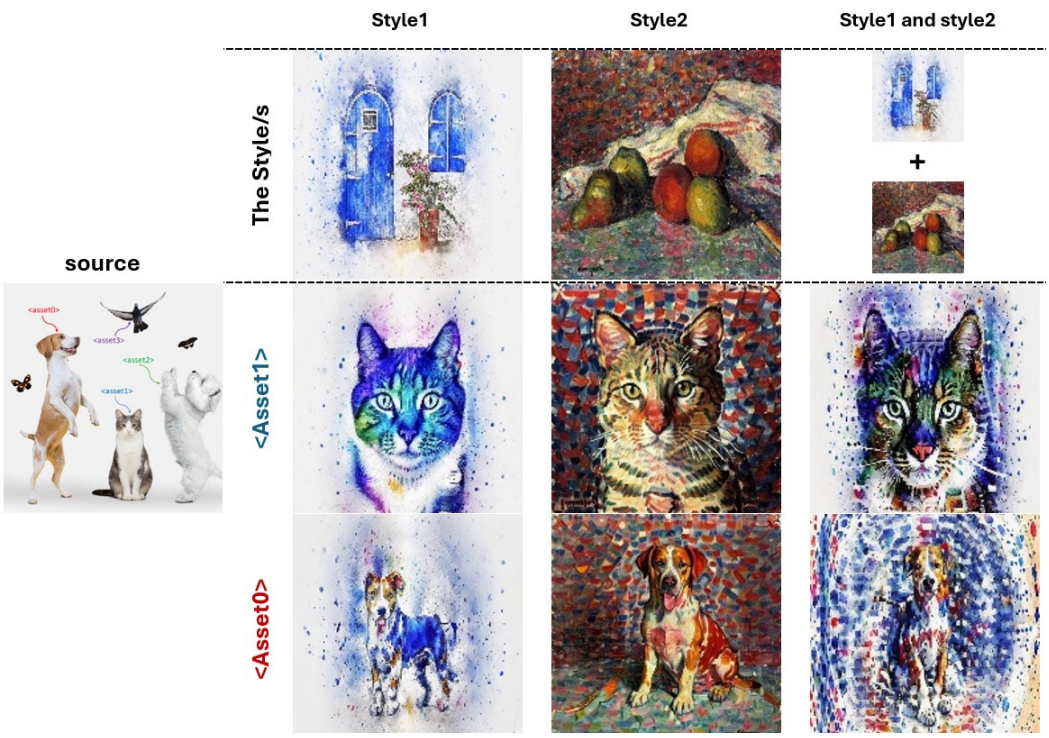

Fig. 6. Results with multiple styles. After training the model with two different styles, the first and second columns show generated images using a single style per column, while the third column demonstrates the combination of both styles in a single image.

Fig. 7. Results with mixed styles. Different pairs of styles were applied using the same textual prompt, resulting in generated images that mix the trained styles to create a new, mixed style.

At 800 timesteps, the generated images achieve an optimal painting style that

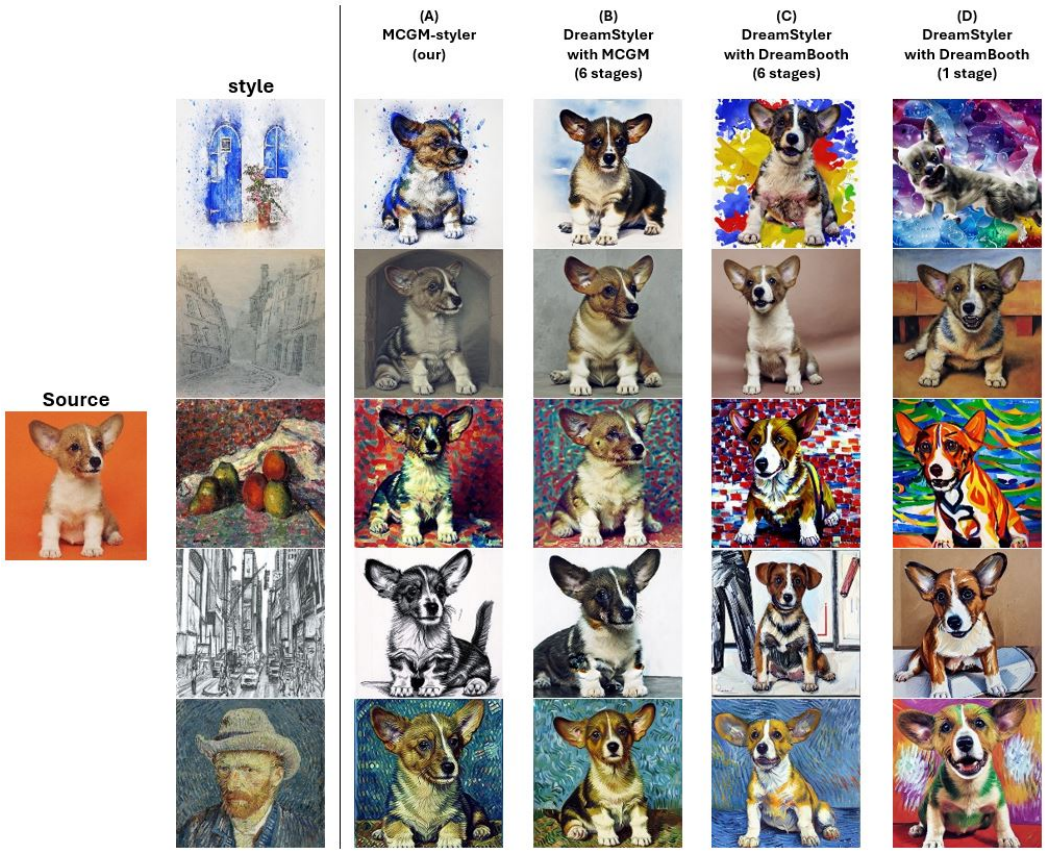

Fig. 8. Comparison between our model and different combination of DreamStyler and DreamBooth. The same sentence was used in all the experiments: *"a painting of* $< asset0 >$ *in the style of* $< s0 >$*."* Here, $< asset0 >$ represents the dog in the source image, while $< s0 >$ corresponds to the style image used in each experiment. (A) represents the results employing our full model for both learning the concept and the styles. (B) and (C) both employ Dreamstyler for the style and MCGM and Dreambooth for the concept, respectivelly. In this case six style training stages are used. Finally, in (D) DreamStyler with a single style training stage in combination with DreamBooth is presented.

closely matches the reference. However, beyond 800 timesteps, the model begins to overfit, prioritizing the style over the trained concept. This overfitting is influenced by the initial concept training phase. Therefore, we selected 800 as the optimal number of timesteps for style training, ensuring a balance between style and concept by matching the timesteps used in both training phases.

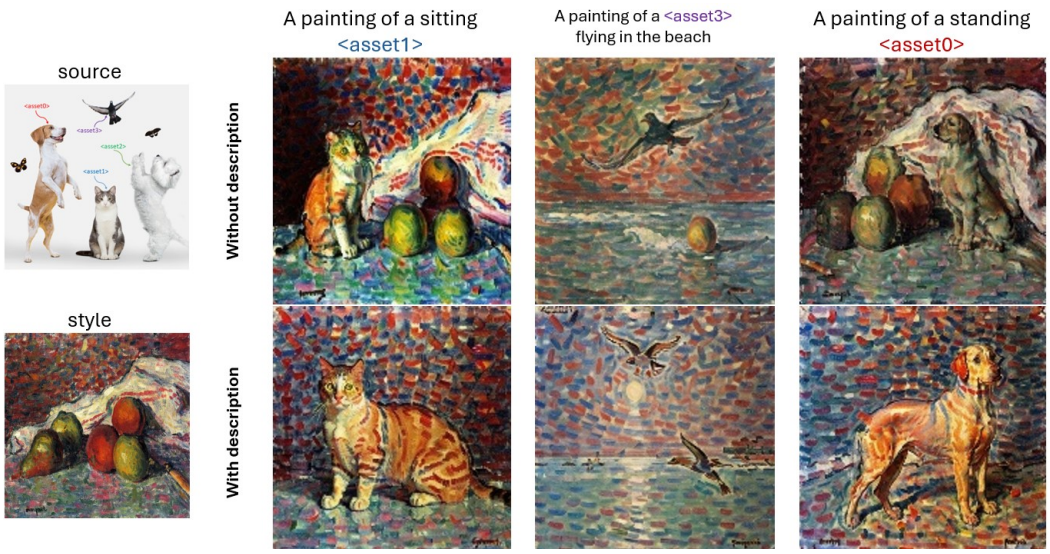

Fig. 9. Different scenes without and with the human description technique. On the left, the source and style used are displayed. The top row shows the different prompts applied. The middle row presents three different concept scenes generated using various text prompts without the human description technique. The bottom row shows the results obtained incorporating human description during training.

4.4 Style-guided text-to-image generation.

MCGM-Styler can also be used to generate images with specific styles without learning specific concepts, extending the capabilities of text-to-image generative models. In this application, we employ generic objects (*e.g.*, a bridge, a house, etc.) rather than specific concepts from the source image to demonstrate the effectiveness of our system in injecting the desired style into the generated images. As illustrated in Fig. 12, we provide a qualitative comparison with previous style-guided approaches.

4.5 Quantitative Evaluation

For the evaluation, we compared our model with DreamStyler under various training conditions like the ones presented in Fig. 8. As detailed in the results in Table 1, we analyzed DreamStyler trained with DreamBooth in six stages, DreamStyler trained using MCGM, and DreamStyler trained with DreamBooth in a single stage to allow direct comparison with our single stage MCGMStyler.

For each of the scenarios mentioned above, we generated 125 images, with 25 samples per style in five different styles. We then computed the *DINO similarity matrix* [64] and *CLIP-I score* [65] for each image and calculated the *average similarity* for each set of 25 images per style. Finally, we derived the *overall average similarity score* for each scenario.

Fig. 10. Contribution of the mask encoder: two experiments using the same source image, target pose mask, text prompt, and target style. The example to the right, without the mask encoder, fails to apply the target pose correctly.

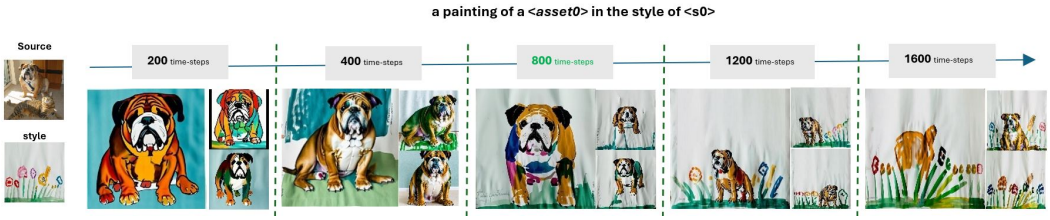

Fig. 11. Experiments with different number of timesteps. For each experiments, three examples are shown. The most accurate results were achieved at 800 timesteps.

The findings indicate that MCGMStyler outperforms the other methods in style transfer quality.

4.6 User Study

To assess the effectiveness of our approach, we conducted a user study involving 25 participants (age 21–38; mixed academic and non-academic backgrounds), to evaluate the quality of pose and style transfer. The study included 25 questions, structured into three parts: (1) *Style Transfer Quality & Perception* (10 questions), where participants rated the realism and fidelity of stylized outputs given a specific target style; (2) *Comparisons & Preferences* (10 questions), comprising object-level and scene-level evaluations across different styles, in which our method was compared to DreamStyler under various training setups as detailed in Table 1; and

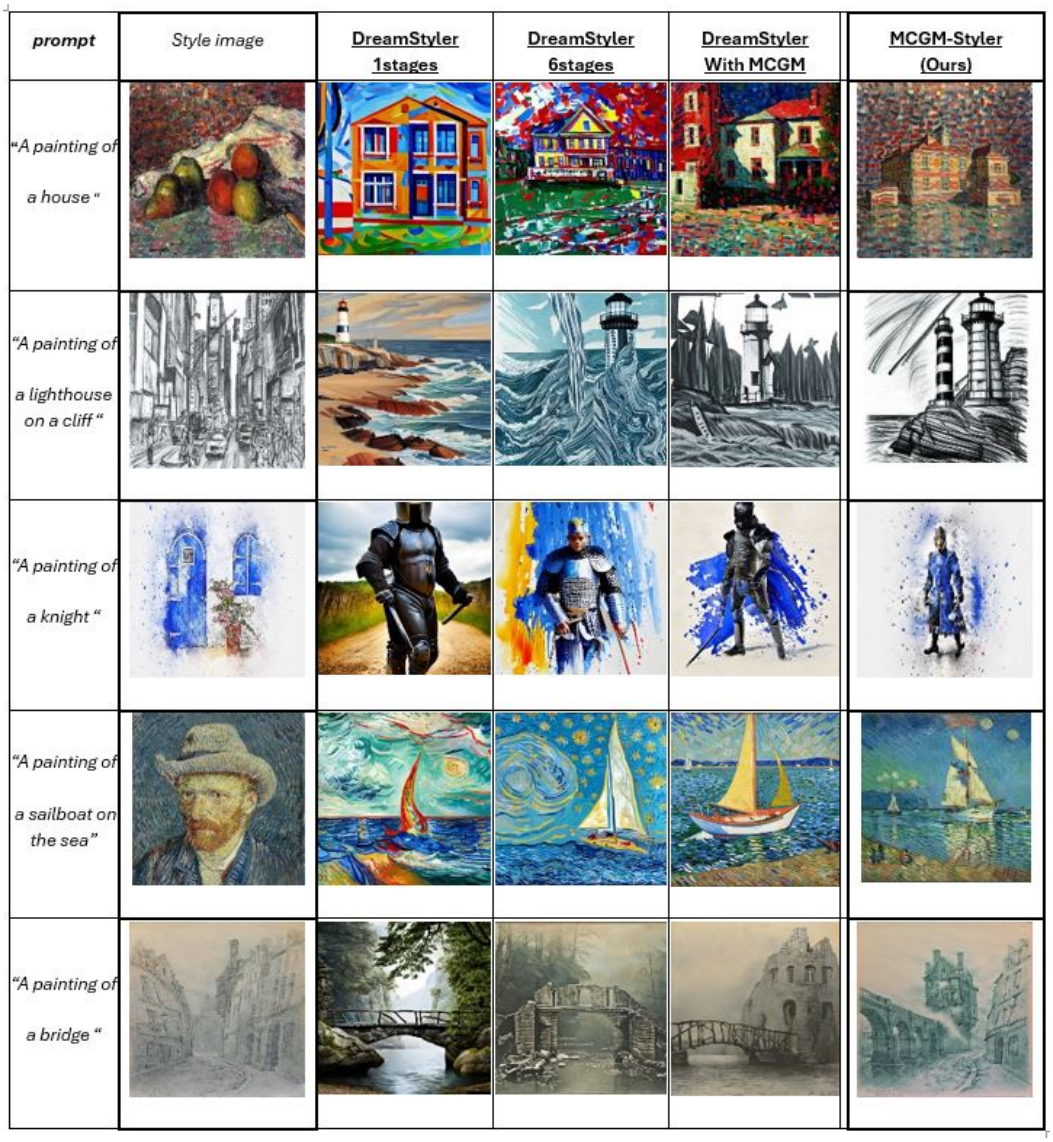

Fig. 12. Style-guided text-to-image generation: Comparison between our model and various configurations of DreamStyler model. The left column shows the input text prompt and target style, the right column displays results from our model, and the center presents DreamStyler outputs under different training setups, including one-step, six-step, and MCGM-pretrained variants.

(3) *Pose and Style Transfer Consistency* (5 questions), where participants assessed the consistency of pose and style across scenes generated from a source image with multiple objects, a target position mask, a target style, and a guiding text prompt.

Table 1. Quantitative Evaluation: Comparison of our model, MCGMStyler, with DreamStyler under different training conditions. **Bold**: best, underline: second best.

Method	DINO ↑ (cosine similarity)	DINO ↓ (Euclidean Distance)	CLIP-I ↑ (Cosine Similarity)
DreamStyler (1 stage)	0.075	38.5	0.591
DreamStyler (6 stage)	0.203	36.1	0.609
DreamStyler (with MCGM)	0.207	35.4	0.643
MCGMStyler	**0.436**	**29.6**	**0.708**

The results showed a strong percentage 77.5% of positive responses (scale 4: 33.33%, and scale 5: 44.16%) in Part 1 and 68.8% (scale 4: 36.45%, and scale 5: 32.33%) in Part 3, indicating high satisfaction with both the fidelity of the style and the integrated transfer of the pose-style. In Part 2, participants expressed a clear preference for our model (MCGM-styler) over three baseline methods, with a selection rate of our method of 65% for object-level tasks and 61.68% for overall scene-level evaluations. These findings support the efficacy of our multi-conditional framework, which enables users to generate high-quality, controllable visual outputs with greater flexibility.

5 Limitations and Future Work

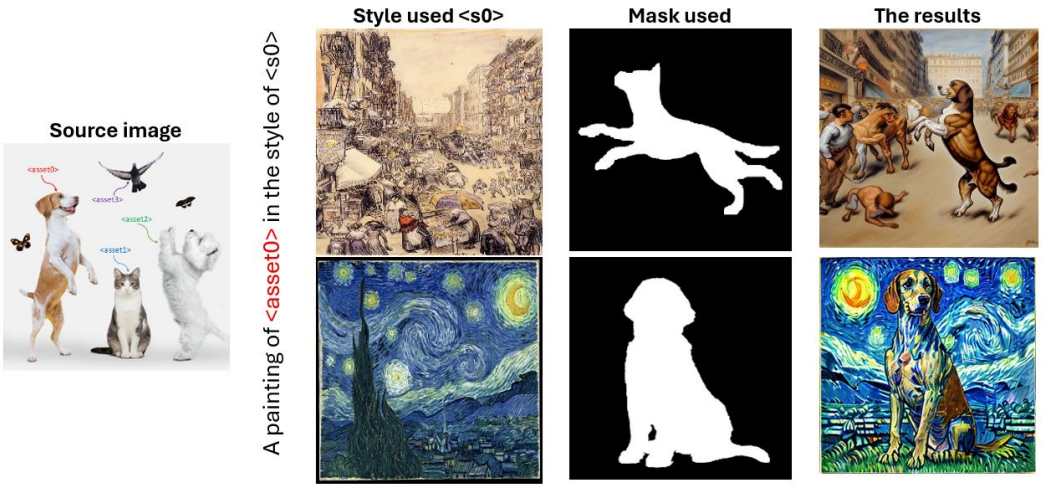

Fig. 13. Decoupling limitation. Two examples of failure cases caused by complex multi-subject interactions in the style image. The left column shows the source image and prompt, the middle column shows the mask and style, and the right column shows the generated results , where the content is entangled with the source style image.

While MCGM-Styler effectively combines pose and style control, it exhibits some limitations. The model can struggle to apply the correct target poses when

generating images with more than three subjects, and as illustrated in Fig. 13, it sometimes fails to decouple content from the style image when highly abstract styles or complex multi-subject interactions with occlusion are involved. Future extensions may include automated style captioning.

Future research will focus on addressing these limitations by extending pose control to support multiple concepts across more than three distinct pose masks. Moreover, we will investigate strategies to enhance the model's capacity to interpret longer prompts, thereby enabling the generation of more detailed and coherent scenes. Additional directions include automated style captioning to improve usability and generalization.

6 Conclusions

In this paper, we proposed MCGM-Styler, a novel generative model that extends MCGM by incorporating style conditioning into the text-to-image synthesis process. By introducing an additional training step, our model effectively learns artistic styles while maintaining precise pose control through masked conditions. Unlike existing methods, MCGM-Styler can generate high-quality images from a single training image, supporting both single- and multi-subject scenarios. Our comparative evaluation with DreamStyler and other works shows that our approach outperforms existing style transfer techniques, demonstrating superior alignment between pose, style, and scene composition. This work paves the way for more flexible and efficient generative models.

References

1. R. Skaik, L. Rossi, T. Fontanini, and A. Prati, "Mcgm: Mask conditional text-to-image generative model," *arXiv preprint arXiv:2410.00483*, 2024. 1, 2, 3, 4, 5, 6, 7, 12
2. N. Ahn, J. Lee, C. Lee, K. Kim, D. Kim, S.-H. Nam, and K. Hong, "Dreamstyler: Paint by style inversion with text-to-image diffusion models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 674–681, 2024. 2, 5, 9, 12
3. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014. 2
4. A. Kamil and T. Shaikh, "Literature review of generative models for image-to-image translation problems," in *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, pp. 340–345, IEEE, 2019. 2
5. D. Liu, J. Zhang, A.-D. Dinh, E. Park, S. Zhang, and C. Xu, "Generative physical ai in vision: A survey," *arXiv preprint arXiv:2501.10928*, 2025. 2
6. T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020. 2
7. J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of naacL-HLT*, vol. 1, Minneapolis, Minnesota, 2019. 2

8. M. Božić and M. Horvat, "A survey of deep learning audio generation methods," *arXiv preprint arXiv:2406.00146*, 2024. 2
9. S. Vasquez and M. Lewis, "Melnet: A generative model for audio in the frequency domain," *arXiv preprint arXiv:1906.01083*, 2019. 2
10. S. Reed *et al.*, "Generative adversarial text to image synthesis," in *International Conference on Machine Learning*, 2016. 2
11. J. Betker, G. Goh, L. Jing, T. Brooks, J. Wang, L. Li, L. Ouyang, J. Zhuang, J. Lee, Y. Guo, *et al.*, "Improving image generation with better captions," *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, vol. 2, no. 3, p. 8, 2023. 2
12. K. Lin, Z. Yang, L. Li, J. Wang, and L. Wang, "Designbench: Exploring and benchmarking dall-e 3 for imagining visual design," *arXiv preprint arXiv:2310.15144*, 2023. 2
13. N. Zhang and H. Tang, "Text-to-image synthesis: A decade survey," *arXiv preprint arXiv:2411.16164*, 2024. 2
14. C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, and M. Norouzi, "Palette: Image-to-image diffusion models," in *ACM SIGGRAPH 2022 conference proceedings*, pp. 1–10, 2022. 2
15. Shelf.io, "The panoramic guide to diffusion models in machine learning," *Shelf.io Blog*, 2023. 2
16. N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22500–22510, 2023. 2, 3, 5
17. A. Goyal and et al., "Break-a-scene: Compositional text-to-image generation via mask conditioning," *arXiv preprint arXiv:2306.13456*, 2023. 2, 3, 4, 5, 6, 7
18. R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-Or, "An image is worth one word: Personalizing text-to-image generation using textual inversion," *arXiv preprint arXiv:2208.01618*, 2022. 2, 3, 5
19. Y. Zuo, J. Xiao, K.-C. Chan, R. Dong, C. Yang, Z. He, H. Xie, and K.-M. Lam, "Towards multi-view consistent style transfer with one-step diffusion via vision conditioning," *arXiv preprint arXiv:2411.10130*, 2024. 3
20. F. He, G. Li, M. Zhang, L. Yan, L. Si, F. Li, and L. Shen, "Freestyle: Free lunch for text-guided style transfer using diffusion models," *arXiv preprint arXiv:2401.15636*, 2024. 3
21. S. Doddapaneni, K. Sayana, A. Jash, S. Sodhi, and D. Kuzmin, "User embedding model for personalized language prompting," *arXiv preprint arXiv:2401.04858*, 2024. 3
22. A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, *et al.*, "Learning transferable visual models from natural language supervision," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2021. 3
23. A. Ramesh, M. Pavlov, G. Goh, S. Gray, *et al.*, "Hierarchical text-conditional image generation with clip latents." https://openai.com/dall-e-2, 2022. 3
24. T. Zhang and H. Tang, "Style transfer: A decade survey," *arXiv preprint arXiv:2506.19278*, 2025. 3
25. K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *arXiv preprint arXiv:1506.05517*, 2015. 3
26. M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014. 3
27. A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," *arXiv preprint arXiv:1610.09585*, 2017. 4
28. P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *arXiv preprint arXiv:1611.07004*, 2017. 4
29. T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "Attngan: Fine-grained text-to-image generation with attentional generative adversarial networks," *arXiv preprint arXiv:1711.10485*, 2018. 4

30. N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher, "Ctrl: A conditional transformer language model for controllable generation," *arXiv preprint arXiv:1909.05858*, 2019. 4
31. T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *arXiv preprint arXiv:1812.04948*, 2019. 4
32. A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," *arXiv preprint arXiv:2102.12092*, 2021. 4
33. R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022. 4
34. L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023. 4
35. Y. Endo, "Masked-attention diffusion guidance for spatially controlling text-to-image generation," *The Visual Computer*, 2023. 4
36. Y. Kawano and Y. Aoki, "Maskdiffusion: Exploiting pre-trained diffusion models for semantic segmentation," *arXiv preprint arXiv:2403.11194*, 2024. 4
37. M.-Y. Chang and et al., "Muse: Text-to-image generative models with mask guidance and diffusion mechanisms," *arXiv preprint arXiv:2305.18254*, 2023. 4
38. G. Couairon, R. Beaumont, L. Sigal, and A. Alahi, "Diffedit: Diffusion-based semantic image editing with mask guidance," *arXiv preprint arXiv:2305.10855*, 2023. 4
39. J. Lee, S. Jang, J. Jo, J. Yoon, Y. Kim, J.-H. Kim, J.-W. Ha, and S. J. Hwang, "Text-conditioned sampling framework for text-to-image generation with masked generative models," *arXiv preprint arXiv:2304.01515*, 2023. 4
40. D. Bashkirova, Y. Li, A. Shrivastava, A. A. Efros, and E. Shechtman, "Masksketch: Unpaired structure-guided masked image generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1234–1243, 2023. 4
41. Y. Zhang, Y. Li, R. Zhang, J.-Y. Zhu, E. Shechtman, and T. Zhang, "Text-guided shape-free object inpainting with diffusion model," *arXiv preprint arXiv:2303.08137*, 2023. 4
42. Y. Li, G.-J. Qi, and J. Luo, "Adversarial learning with mask reconstruction for text-guided image inpainting," *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 1234–1243, 2021. 4
43. C. Wang, X. Li, L. Qi, H. Ding, Y. Tong, and M.-H. Yang, "Semflow: Binding semantic segmentation and image synthesis via rectified flow," *Advances in Neural Information Processing Systems*, vol. 37, pp. 138981–139001, 2024. 4
44. J.-Y. Lee, K.-I. Park, D. Kim, S. Woo, S. J. Park, J. Choo, and N. Kwak, "Maskgan: Towards diverse and interactive facial image manipulation," *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 4
45. P. Zhu, R. Abdal, Y. Qin, and P. Wonka, "Sean: Image synthesis with semantic region-adaptive normalization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5104–5113, 2020. 4
46. T. Fontanini, C. Ferrari, G. Lisanti, M. Bertozzi, and A. Prati, "Semantic image synthesis via class-adaptive cross-attention," *IEEE Access*, vol. 13, pp. 10326–10339, 2025. 4
47. T. Fontanini, C. Ferrari, G. Lisanti, L. Galteri, S. Berretti, M. Bertozzi, and A. Prati, "Frankenmask: Manipulating semantic masks with transformers for face parts editing," *Pattern Recognition Letters*, vol. 176, pp. 14–20, 2023. 4
48. A. Ergasti, C. Ferrari, T. Fontanini, M. Bertozzi, and A. Prati, "Controllable face synthesis with semantic latent diffusion models," in *Pattern Recognition. ICPR 2024 International Workshops and Challenges* (S. Palaiahnakote, S. Schuckers, J.-M. Ogier, P. Bhattacharya, U. Pal, and S. Bhattacharya, eds.), (Cham), pp. 337–352, Springer Nature Switzerland, 2025. 4

49. H. Wang, P. Xing, R. Huang, H. Ai, Q. Wang, and X. Bai, "Instantstyle-plus: Style transfer with content-preserving in text-to-image generation," *arXiv preprint arXiv:2407.00788*, 2024. 4, 5
50. K. Sohn, N. Ruiz, K. Lee, D. C. Chin, I. Blok, H. Chang, J. Barber, L. Jiang, G. Entis, Y. Li, *et al.*, "Styledrop: Text-to-image generation in any style," *arXiv preprint arXiv:2306.00983*, 2023. 4, 5
51. G. Zhang, K. Sohn, M. Hahn, H. Shi, and I. Essa, "Finestyle: Fine-grained controllable style personalization for text-to-image models," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 4
52. D.-Y. Chen, H. Tennent, and C.-W. Hsu, "Artadapter: Text-to-image style transfer using multi-level style encoder and explicit adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8619–8628, 2024. 4
53. P. Xing, H. Wang, Y. Sun, Q. Wang, X. Bai, H. Ai, R. Huang, and Z. Li, "Csgo: Content-style composition in text-to-image generation," *arXiv preprint arXiv:2408.16766*, 2024. 4
54. M. Lei, X. Song, B. Zhu, H. Wang, and C. Zhang, "Stylestudio: Text-driven style transfer with selective control of style elements," *arXiv preprint arXiv:2412.08503*, 2024. 4
55. C. Liu, V. Shah, A. Cui, and S. Lazebnik, "Unziplora: Separating content and style from a single image," *arXiv preprint arXiv:2412.04465*, 2024. 5
56. F. Botti, A. Ergasti, L. Rossi, T. Fontanini, C. Ferrari, M. Bertozzi, and A. Prati, "Mamba-st: State space model for efficient style transfer," in *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pp. 7786–7795, February 2025. 5
57. L. Han, Y. Li, H. Zhang, P. Milanfar, D. Metaxas, and F. Yang, "Svdiff: Compact parameter space for diffusion fine-tuning," *arXiv preprint arXiv:2303.11305*, 2023. 5
58. N. Kumari, B. Zhang, R. Zhang, E. Shechtman, and J.-Y. Zhu, "Multi-concept customization of text-to-image diffusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 5
59. Y. Wei, Y. Zhang, Z. Ji, J. Bai, L. Zhang, and W. Zuo, "Elite: Encoding visual concepts into textual embeddings for customized text-to-image generation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15943–15953, 2023. 5
60. R. Gal, M. Arar, Y. Atzmon, A. H. Bermano, G. Chechik, and D. Cohen-Or, "Encoder-based domain tuning for fast personalization of text-to-image models," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1–13, 2023. 5
61. E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," in *International Conference on Learning Representations (ICLR)*, 2022. 5
62. S. Ryu, "Low-rank adaptation for fast text-to-image diffusion fine-tuning," *Low-rank adaptation for fast text-to-image diffusion fine-tuning*, 2023. 7, 8
63. J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *Proceedings of the 40th International Conference on Machine Learning* (A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, eds.), vol. 202 of *Proceedings of Machine Learning Research*, pp. 19730–19742, PMLR, 23–29 Jul 2023. 9
64. S. Amir, Y. Gandelsman, S. Bagon, and T. Dekel, "Deep vit features as dense visual descriptors," *arXiv preprint arXiv:2112.05814*, vol. 2, no. 3, p. 4, 2021. 16
65. G. Kwon and J. C. Ye, "Clipstyler: Image style transfer with a single text condition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18062–18071, 2022. 16