Proceedings of the 4th International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/2026.39338

A Review of Visual Style Transfer Technologies and Applications

Zhiyue Li

The Second High School Attached to BNU, Beijing, China
3975484628@qq.com

Abstract. In recent years, generative artificial intelligence has grown fast and pushed visual style transfer toward better control, more modes and automatic creation. This paper looks back at how these technologies have grown and at how people use them. It first explains GAN-based methods and CycleGAN that turn one image into another picture. Then it talks about newer work built on diffusion models, vision-language models, large language models and AI Agents. The move of style transfer from still pictures to video generation is also studied here. It also lists the main problems, such as fine-grained control, content-style balance, time consistency, computing cost, and copyright and ethics. On the whole, the field is moving from image translation to a full system for visual creation. Later work will probably care more about controllable generation, multimodal interaction, reliable testing, and human-AI creation.

Keywords: visual style transfer, GAN, CycleGAN, generative artificial intelligence

1. Introduction

In recent years, artificial intelligence has changed a great deal. It has moved from "understanding visual content" to "making visual content." Early computer vision work mostly studied simple jobs such as image classification and object detection. Its main goal then was to help computers read the information inside pictures people already had. Later, deep learning grew quickly, and so did generative models and large-scale pretraining. Generative AI then appeared, and machines became able to make new visual content. They could build it from a picture or from a short piece of written text. Because of this, visual content generation has become an important research area. It is now used in digital art, film and TV production, advertising design, games and social media.

Visual style transfer is an important branch of visual content generation. Its goal is to change how an image looks while keeping its content and meaning structure safe. Here, "style" covers things such as color spread, texture, lines and brushstrokes, while "content" means the people, objects, scenes and the way they are placed in space. For example, this technology can turn a photo into an oil-painting, a sketch or an anime picture, and the people and scenes in it stay almost the same. Compared with common image-generation tasks, its key problem is how to pull apart, show and then join again the content features and the style features of one picture.

Early methods used Convolutional Neural Networks, or CNNs, to pull out features for style work. They used deep networks to show content features and style features one by one. In this way a

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.39338

picture could be rebuilt in a new style. Later, Generative Adversarial Networks, or GANs, learned how real images are spread out. They did this through adversarial training between a generator and a discriminator, and it helped make high-quality images [1]. Models such as CycleGAN went further. They dealt with turning one image into another when no paired training data existed [2]. In recent years, large multimodal models, AI Agents and automatic video generation have all grown. So style transfer has moved beyond single images. It now also covers text-controlled generation, automatic content creation, and video transfer that keeps time continuity.

This paper does not train any model or run any experiment. Instead, it looks at visual style transfer from the sides of technical principles and application growth. First, it explains the basic structure and the adversarial training mechanism of GANs. It also studies the idea of cycle consistency in CycleGAN and its strong points in unpaired image-to-image translation. Then it looks at how style transfer is linked to large language models, or LLMs [3], multimodal models and AI Agents. It studies how natural language can be an important way for people to control visual style and content generation. Last of all, it looks at the growth of automatic video generation and at the trend of moving style transfer from still pictures to moving visual content.

2. Visual style transfer and applications

Visual style transfer is a technique that changes how an image looks while keeping its content and meaning structure. As said above, one can describe a picture in terms of content and style. Content covers the people, objects, scenes and the space structure, while style shows up in things such as colors, textures, lines and brushstrokes. The goal of style transfer is to put the style features of one picture onto another picture. For example, a photo of a city can be turned into an oil-painting, a sketch or an anime picture, and the buildings, roads and people in it stay almost the same. So style transfer is not just a matter of changing colors. It also needs a model to learn the features of different styles and to find a good balance between keeping the content and changing the style.

The most common uses of visual style transfer are artistic image generation and image-to-image translation. It makes professional visual creation easier for ordinary people to try. It can work between photos and anime pictures, between day and night, between different seasons and weather, and between real and virtual scenes. It has been used in advertising design, game development, film and TV production and social media content creation. As the need for such uses has grown, old methods built on convolutional neural networks have shown their limits in quality and in flexibility. Generative models such as GANs have made the results look more real by learning how images in different domains are spread out. Among them, CycleGAN can move between domains without paired training data, and this has greatly widened the real uses of style transfer.

In recent years, visual style transfer has moved further out from still pictures into multimodal generation, smart content creation and video generation. Large language models and multimodal models have grown, so users no longer need to give a reference image. They can now describe the visual effect they want with words alone. AI Agents can also put the steps of image generation, style choice, content change and result improvement into order, and this forms an automatic creative workflow. In video generation, style transfer has to do more than change how each single frame looks. It also has to keep time consistency between frames, so as to cut down flicker and sudden changes in how objects look. As a result, visual style transfer is growing from an old artistic picture tool into an important technology that links image generation, multimodal models, smart agents and video generation.

DOI: 10.54254/2755-2721/2026.39338

3. Generative Adversarial Networks

A Generative Adversarial Network, or GAN, is a deep generative model that makes new samples through implicit distribution modeling [1]. It first takes latent variables z from a simple prior distribution $p_z(z)$. Then it uses a generator with parameters θ_G to map them into the image space, which gives $\tilde{x} = G(z;\theta_G),\ z \sim p_z(z)$. Some methods try to work out probability densities, but a GAN works in another way. It uses the adversarial feedback from a discriminator, and step by step it brings the generated distribution p_g nearer to the real data distribution p_{data}. For picture data, the generator can encode patterns such as textures, colors, edges, shapes and space structures through many nonlinear changes. This builds continuous representations in the latent space. So after enough training, the model can build new pictures from random variables, and these pictures look like real ones in both small details and the whole layout. In plain words, this process learns the spread of the training data and then samples from it again.

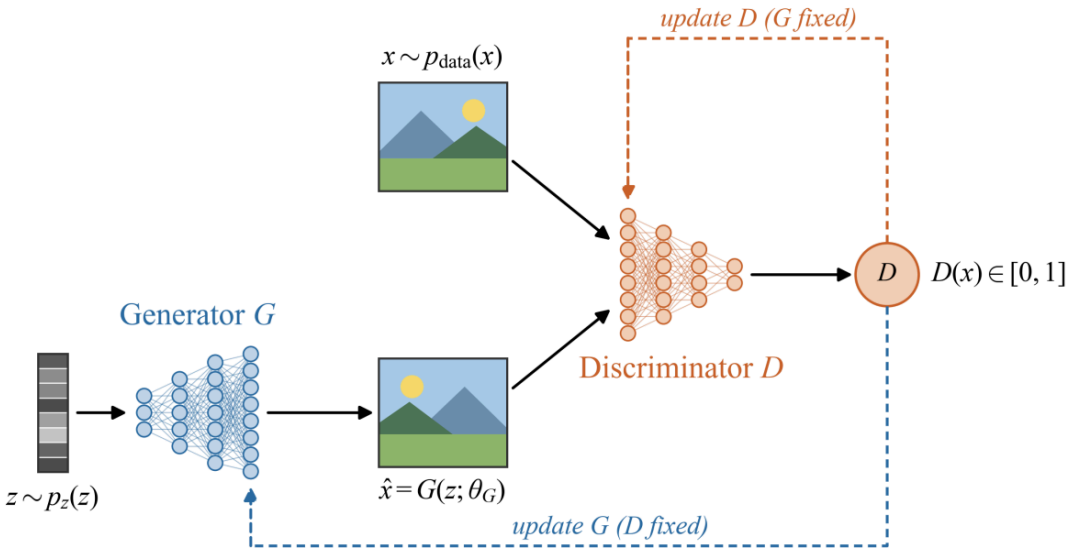

Figure 1. Architecture of GAN

As Figure 1 shows, a GAN has two neural networks that work against each other, namely a generator G and a discriminatorD. The generator takes in random noise or conditional information, and it tries to make fake samples that fool the discriminator. The discriminator takes in both real samples x and generated samples $G(z)$. It gives out the probability $D(x) \in [0,1]$ that an input comes from the real data distribution. During training, the two sets of parameters are usually updated in turn. First the generator is kept fixed so the discriminator can learn to tell real samples from fake ones. Then the discriminator is kept fixed so the generator can learn to fool it better. In the best case, when $p_g = p_{data}$, the discriminator gives $D(x) = 1/2$ to any sample. This means the generator has learned to sample from a distribution that matches the real one. In real use, people often pick the non-saturating generator loss, because it makes the gradient signal stronger early in training. Through this contest, the model lifts both generation quality and judging accuracy by passing errors back, and no rules for textures or structures are written by hand. The whole objective of the classical GAN can be written as follows:

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z)))] \quad (1)$$

DOI: 10.54254/2755-2721/2026.39338

GANs can do more than copy the surface look of one category. They can also learn the common structures and style differences between image domains from conditional inputs and feature limits. In image generation without conditions, latent variables can control pose, layout and small details. In image restoration, the masked parts and the areas around them act as conditions, so the model can fill in the missing content by looking at nearby textures and high-level meaning. In style transfer, the source image x gives the content structure, while the style encoding s sets the colors, brushstrokes, textures and brightness of the target domain, and the generator then makes the target image $\hat{y}$. During training, one can join the adversarial term L_{adv}, the content-preservation term $L_{content}$ and the style-matching term L_{style} by weighted summation. The first one makes the result look more real in the target domain, while the other two stop the meaning from changing and keep the style statistics steady. In this way, image generation, image restoration and visual style transfer can all be seen as problems of learning a conditional distribution, and a common objective can be written as follows:

$$\hat{y} = G(x, s), \quad L_G = L_{adv} + \lambda_c L_{content} + \lambda_s L_{style} \tag{2}$$

Adversarial loss puts direct limits on high-frequency details and on the whole data distribution, and GAN pictures gain a lot from this. So the pictures they make usually have clear textures, natural colors and a strong sense of being real. But three big problems still hold them back. First, the generator and the discriminator form a non-convex dynamic game. If one side is updated too fast, gradients may vanish, or the training may swing back and forth and never settle down. Picture quality can then jump up and down during training. Second, the first GANs were driven mostly by random noise, and latent variables had no clear link to meaning. So it was hard to control object categories, space layouts and style strength in an exact way. Third, the model may fall into mode collapse, where different latent variables map to almost the same samples. This gives $G(z_i) \approx G(z_j),\ z_i \neq z_j$ and stops the generated distribution from covering the many modes of the real one. To fix these weak points, people can use Wasserstein distance, gradient penalty and spectral normalization to make training more stable. They can also use conditional labels, attention and latent-variable disentanglement to control the model better. Feature matching, minibatch discrimination and diversity regularization can ease it. Even so, they still force a choice between fidelity, diversity and computing cost. So stable and clear style transfer is still an important direction in GAN research.

4. CycleGAN and unpaired image translation

A normal GAN takes random noise as its input, and it works well for making new samples. But it has no way to keep the content structure of an input picture. Even if one puts an image in instead of noise, a single adversarial loss only pushes the output to fit the target domain distribution, and it cannot make sure the original content stays while textures or colors change. Old methods can reach pixel-level supervision by using paired data, but paired samples are hard to get and cost a lot for jobs such as artistic style transfer and season change. So unpaired image translation methods came up, and CycleGAN is a well-known example of them [2].

DOI: 10.54254/2755-2721/2026.39338

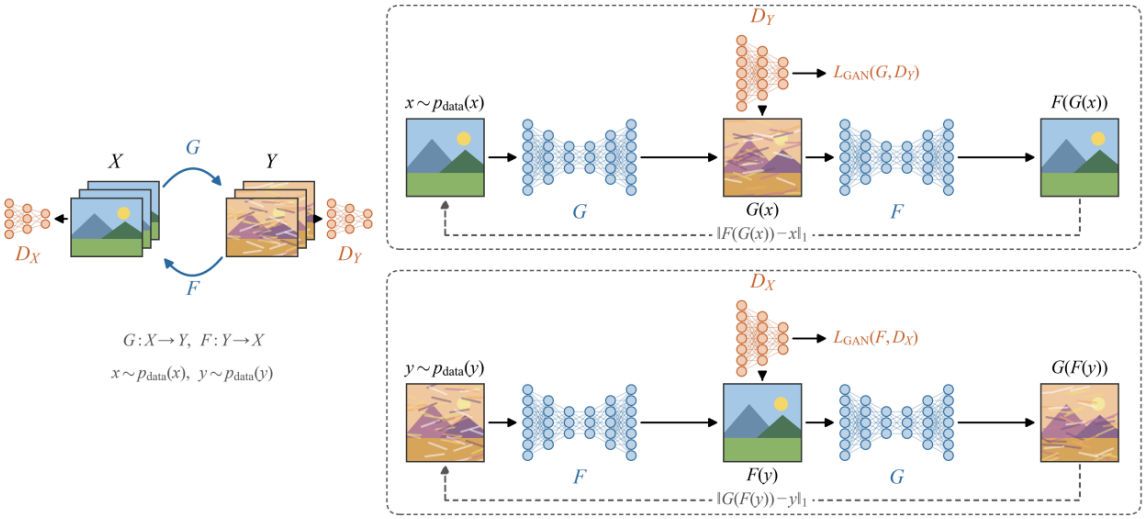

Figure 2. Architecture of CycleGAN

As Figure 2 shows, let X and Y stand for the source image domain and the target image domain. The training samples x and y are taken on their own from the data distributions $p_{data}(x)$ and $p_{data}(y)$, and they do not have to be paired. CycleGAN has two generators, G and F, and it also has two discriminators, D_Y and D_X. The generator G maps pictures from the source domain to the target domain, while the generator F does the mapping in the other direction. The discriminator D_Y tells real target-domain pictures y from generated ones D_X, and the discriminator D_X tells real source-domain pictures x from generated ones $F(y)$. The mapping relations above can be written as follows:

$$x \sim p_{data}(x),\ \ y \sim p_{data}(y),$$

$$G : X \to Y,\ \ F : Y \to X, \tag{3}$$

So the model builds two closed conversion paths, $X \to Y \to X$ and $Y \to X \to Y$.

The core objective of CycleGAN has two parts, namely bidirectional adversarial losses and a cycle-consistency loss. In the $X \to Y$ direction, the generator G tries to make pictures that can fool D_Y, while the discriminator D_Y tries to get better at telling real pictures from generated ones. When one uses the logarithmic form of the first GAN objective, the adversarial loss in this direction is defined as

$$L_{GAN}(G, D_Y) = E_y[\log D_Y(y)] + +E_x[\log(1 - D_Y(G(x)))] \tag{4}$$

Likewise, the adversarial loss in the Y→X direction is

$$L_{GAN}(F, D_X) = E_x[\log D_X(x)] \ + E_y[\log(1 - D_X(F(y)))] \tag{5}$$

The two adversarial losses push the generated results toward the data distributions of the target domain and of the source domain. But matching at the distribution level cannot promise that the content of one single input picture stays the same. To deal with this problem, CycleGAN adds a cycle-consistency constraint. After a picture is mapped in the forward direction, the inverse

DOI: 10.54254/2755-2721/2026.39338

generator brings it back to its first domain, and the rebuilt result should stay as close to the first input as it can. Its loss function is written as

$$L_{cyc}(G,F) = E_x[\|F(G(x)) - x\|_1] + E_y[\|G(F(y)) - y\|_1] \quad (6)$$

where the L1 norm measures the absolute error between the rebuilt picture and the first picture at the pixel level.

When there are no other regularization terms, the whole objective function of CycleGAN is

$$L = L_{GAN}(G, D_Y) + L_{GAN}(F, D_X) + \lambda L_{cyc}(G, F) \quad (7)$$

The parameter λ adjusts the weight of the cycle-consistency loss inside the whole objective. When λ is small, the model cares more about how real the target-domain style looks, but it gets weaker at keeping content. When λ is very large, the generator pays more attention to rebuilding the input picture, and this limits how much the style can change. So λ has to be set from the differences between the datasets and from the goals of the task. Model training can be written as the following minimax problem. The generators G and F make the whole objective as small as they can, while the discriminators D_X and D_Y make the adversarial objective as large as they can:

$$G^*, F^* = argmin_{G,F} max_{D_X, D_Y} L(G, F, D_X, D_Y) \quad (8)$$

The main strong point of CycleGAN is that its training does not need pixel-level paired picture samples. It only needs data collected from the two image domains on their own, and from that it can learn the mappings in both directions. This feature cuts the cost of data collection and image registration. So CycleGAN fits jobs such as artistic style transfer, season and weather change, and appearance change between imaging modalities.

But CycleGAN still has clear weak points. First, adversarial training may suffer from mode collapse, where the generator tends to make a small number of similar results, so the samples do not differ enough. Second, cycle consistency mainly gives a pixel-level rebuilding constraint, and it does little to keep high-level meaning or fine-grained geometric structure. When it works on human faces, text, small objects or complex space relations, the model may make local texture artifacts, change features that show who a person is, twist characters, or deform structures. Last, when moving between image domains needs big changes in the number of objects or in the geometric layout, the idea that the cycle can be run backwards may no longer be true. This limits the quality of the change. So in real use, one should judge first whether the cycle-consistency assumption fits the job, and add perceptual, semantic or structure-preserving losses when needed.

5. Style transfer in the era of generative AI

In the time of generative AI, the base of style transfer has moved from "image-domain mapping" to "diffusion generation controlled by language." CLIP puts pictures and text into one shared meaning space through image-text contrastive learning, so styles can be set with words instead of target-domain datasets. In diffusion models, ControlNet adds a trainable copy of the encoder branch and uses edge maps, depth maps or pose maps to limit the visual structure, so the content of the first picture stays while its style changes [4]. InstructPix2Pix uses GPT-3 to write editing instructions and to make paired training data, and this gives an editing model that can follow an instruction such as "turn it into a watercolor painting" [5]. For getting a style, LoRA puts low-rank matrices into the

DOI: 10.54254/2755-2721/2026.39338

attention layers and can learn a new style from only a few pictures [6]. IP-Adapter, on the other hand, puts in features of a reference picture through disentangled cross-attention, and it allows reference-picture-guided stylization without fine-tuning [7].

The job of LLMs and Agents is to put the models above together into systems that run on natural language. Visual ChatGPT uses a prompt manager to tell ChatGPT what each visual foundation model does and what input and output format it uses. ChatGPT then decides the order in which to call them and passes middle pictures between several rounds of talk [8]. HuggingGPT goes further and sets the work into four stages, namely task planning, model selection, task execution and result summarization, and the LLM picks the right expert models from a model library [9]. To make results better, Idea2Img uses an iterative "generation–self-evaluation–revision" loop. In it, a multimodal large model gives written feedback on the pictures and changes the prompts before new results are made [10].

In video generation, the key technical problem is time consistency. Rerender-A-Video first restyles key frames and then pushes the results to the other frames by using cross-frame limits guided by optical flow [11]. AnimateDiff trains a plug-and-play temporal motion module, and it can join with LoRAs of different styles to make animations in a given style at once [12]. At a higher level, VideoDirectorGPT first uses an LLM to turn text into a "video plan" that holds storyboard descriptions, entity layouts and consistency groupings. Then it drives a video-generation model to make each shot one after another, so characters and styles stay the same across shots [13]. The main limits today are as follows. Words cannot catch small differences in brushstrokes and textures in an exact way, and multimodal models still cannot judge style similarity and beauty in a safe way. Also, calling many models costs much computing power, and keeping consistency in long videos is still not solved.

6. Challenges and future development

Generative AI has moved style transfer forward in a big way, but the technologies we have still run into some basic limits. On the side of control, the sampling process of diffusion models is random in its nature. Words cannot describe small differences in brushstrokes and textures, so users can only control local parts in a rough way. The trade-off between content and style strength still rests on tuning parameters by hand, and a way to control it with numbers is missing. In video work, cross-frame attention and optical-flow limits can ease flicker in short clips. But keeping characters and styles the same over long time spans and across many shots is still an open problem. At the agent level, the "generation–evaluation–revision" loop rests on the judgment of multimodal models, while style transfer has no objective and unified test. Small errors can build up over many rounds, and this brings a large computing cost. A deeper problem is that today's models only learn the statistical spread of visual features. They can copy the surface texture of a style, but they find it hard to catch the cultural background and the creative intent behind a work.

Copyright and ethical problems also cannot be left out. The training data of mainstream models mostly come from the Internet and hold many copyrighted works. Light fine-tuning methods such as LoRA make these arguments sharper, because now a few samples are enough to copy the style of one artist. We must also answer where the training data come from and whether their use is allowed. How to watermark and label generated content, and where the legal line for style imitation should lie, are still open. All these must be answered before the technology can be used in a standard way.

Besides, native multimodal models should help join instruction understanding and image generation more closely, and this will cut the information lost between language and vision. Test models built to judge style similarity and beauty will give agents a safer base for self-correction.

DOI: 10.54254/2755-2721/2026.39338

Keeping consistency in long videos and running the models fast are still key directions for video style transfer. It should be made clear that generative AI should add to human creativity, not take its place. Judging beauty, showing feeling and understanding culture still sit at the center of the artistic value of a work. So human-AI creation will be a more real path in the years to come.

7. Conclusion

This paper looks back at visual style transfer from the sides of its technical principles and of how its uses have grown. GANs have made generated pictures look more real through adversarial training between generators and discriminators, while CycleGAN has dealt with style transfer between unpaired pictures through cycle-consistency limits. Together they have built the base for style transfer that learns from data. In the time of generative AI, visual style transfer is spreading out from changing still pictures to text-controlled image generation, video generation and automatic content creation. A clear split of work is slowly taking shape. LLMs and smart agents take care of understanding intent, planning tasks and controlling the process, while visual generation models make the actual content.

On the whole, the growth of visual style transfer shows how artificial intelligence has moved from recognizing pictures to creating visual content. Today's technologies still have problems with generation accuracy, stability across time and copyright rules. But they still hold much room to grow in human-AI collaboration and in digital creation as multimodal models and agent technologies keep becoming mature.

References

[1] Goodfellow, I. J., et al. (2014). Generative adversarial networks. arXiv. doi: 10.48550/ARXIV.1406.2661.
[2] Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2020, August 24). Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv: arXiv:1703.10593. Accessed: May 28, 2022. [Online]. Available: http://arxiv.org/abs/1703.10593
[3] Liu, R., & Koehn, P. (2026, February 16). Text style transfer with parameter-efficient LLM finetuning and round-trip translation. arXiv: arXiv:2602.15013. doi: 10.48550/arXiv.2602.15013.
[4] Zhang, L., Rao, A., & Agrawala, M. (2023, November 26). Adding conditional control to text-to-image diffusion models. arXiv: arXiv:2302.05543. doi: 10.48550/arXiv.2302.05543.
[5] Brooks, T., Holynski, A., & Efros, A. A. (2023, January 18). InstructPix2Pix: Learning to follow image editing instructions. arXiv: arXiv:2211.09800. doi: 10.48550/arXiv.2211.09800.
[6] Hu, E. J., et al. (2021, October 16). LoRA: Low-rank adaptation of large language models. arXiv:2106.09685. https://doi.org/10.48550/arXiv.2106.09685
[7] Ye, H., Zhang, J., Liu, S., Han, X., & Yang, W. (2023, August 13). IP-adapter: Text compatible image prompt adapter for text-to-image diffusion models. arXiv:2308.06721. https://doi.org/10.48550/arXiv.2308.06721
[8] Wu, C., Yin, S., Qi, W., Wang, X., Tang, Z., & Duan, N. (2023, March 8). Visual ChatGPT: Talking, drawing and editing with visual foundation models. arXiv:2303.04671. https://doi.org/10.48550/arXiv.2303.04671
[9] Shen, Y., Song, K., Tan, X., Li, D., Lu, W., & Zhuang, Y. (2023, December 3). HuggingGPT: Solving AI tasks with ChatGPT and its friends in hugging face. arXiv:2303.17580. https://doi.org/10.48550/arXiv.2303.17580
[10] Yang, Z., et al. (2024, August 14). Idea2Img: Iterative self-refinement with GPT-4V(ision) for automatic image design and generation. arXiv:2310.08541. https://doi.org/10.48550/arXiv.2310.08541
[11] Yang, S., Zhou, Y., Liu, Z., & Loy, C. C. (2023, September 17). Rerender a video: Zero-shot text-guided video-to-video translation. arXiv:2306.07954. https://doi.org/10.48550/arXiv.2306.07954
[12] Guo, Y., et al. (2024, February 8). AnimateDiff: Animate your personalized text-to-image diffusion models without specific tuning. arXiv:2307.04725. https://doi.org/10.48550/arXiv.2307.04725
[13] Lin, H., Zala, A., Cho, J., & Bansal, M. (2024, July 12). VideoDirectorGPT: Consistent multi-scene video generation via LLM-guided planning. arXiv:2309.15091. https://doi.org/10.48550/arXiv.2309.15091