DreamShot: Teaching Cinema Shots to Latent Diffusion Models

DreamShot: Teaching Cinema Shots to Latent Diffusion Models

Tommaso Massaglia[1,*], Bartolomeo Vacchetti[1,†] and Tania Cerquitelli[1]

[1]Polytechnic of Turin, 24 Corso Duca degli Abruzzi, Turin, 10129, Italy

Abstract

In recent years, several text-image synthesis models have been released that are increasingly capable of synthesizing realistic images close to the input. Among the various state-of-the-art techniques and models, the introduction of the open-source latent diffusion model Stable Diffusion [1] has led to significant developments in text-to-image generation in recent months. By using techniques such as DreamBoot [2] and Textual Inversion [3], it is possible to refine further and control the generation process to produce even more specific output than text alone would allow. We test this approach for generating three specific cinematographic shot types: Close-up, Medium Shot, and Long Shot. By fine-tuning based on Stable Diffusion 1.5 using a small dataset of 600 labelled and captioned film frames, we achieve a noticeable increase in CLIP -T and DINO scores and an overall noticeable qualitative improvement (as indicated by our human-run evaluation survey) in image likability, compliance, and shot type correctness.

Keywords

Diffusion Models, Shot Types, text to image

1. Introduction

Image generation has seen a major rise in popularity since the release of the Diffusion Model [4] architecture, with improvements in the quality of the generations that made the pictures ever so close to realistic art pieces and photos. Being able to generate realistic pictures that follow a given textual description through the use of models such as the Latent Diffusion [5] based *Stable Diffusion* [1] opens up a multitude of previously unattainable tasks, which are further improved by the ability to add new subjects in a simple way provided by DreamBooth [2]. By using these two techniques it would be possible to, for example, automatically generate an advertising campaign for a novel product or perform seamless photo editing through textual instructions. Notably, cinema heavily relies on the utilization and creation of reference images to enhance workflow efficiency. With the capacity to generate realistic images, generating expressive reference images that precisely convey the intended shot becomes readily accessible to all, eliminating the need for an extensive reference library or artistic drawing skills. These reference images and sketches are widely employed in **storyboarding**, an essential film-making technique that aids in visualizing the narrative and streamlining the filming process. Within this context, the selection of the desired shot type plays an important role, as it significantly influences the audience's focus and emotions [6].

Table 1
Total number and their respective downloads of the top 100 models hosted on Civitai.

type	number	downloads
DreamBooth Checkpoint	70	5.575.099
Lora DreamBooth	26	1.670.288
Textual Inversion	4	348.187

To the best of our knowledge, the use of text-to-image generation models and their fine-tuning in this context remains widely unexplored. In this paper, we explore the use of DreamBooth [2] (as it is the most widely used fine-tuning approach for pre-trained Latent Diffusion models, as shown in table 1) in adding the knowledge of three specific shot types, close-shot, medium-shot, and long shot, to a pre-trained version of *stable-diffusion-v-1-5* [1]. Given a textual input and a desired shot scale, our methodology is able to generate synthetic scenes that are semantically close to the input and to the scale selected. Using the same testing setup that was proposed in the original DreamBooth [2] paper, we achieve an improvement over the baseline model in both CLIP-T [7] and DINO [8] scores. We complement this testing with a survey conducted on 55 subjects which further shows the qualitative improvements achieved by our approach.

Our contributions are the following: the outlining of a methodological approach to fine-tuning an existing latent diffusion model with state-of-the-art techniques (DreamBooth)to teach a new *style*; the steps necessary to build a training set out of unlabeled movie shots in order to fine-tune a pre-trained model; a set of three fine-tuned models catered towards the generations of three specific shot types: close shot, medium shot, and long shot.

The paper is organized as follows: Section 3 covers the methodology and describes the techniques on which our approach relies; Section 2 discusses the methods exploited in the proposed methodology; Section 4 outlines the testing procedure, metrics used, and relevant results.

2. Related Works

2.1. Storyboarding

In recent years a growing number of studies have focused on the automation of video editing tasks. While these works, such as [9] and [10], achieve impressive performance in the generation of a video, either given as input a textual prompt [10], or a combination of textual prompt and image [9], they focus on the generation of motion and do not take into account the shot type used.

By generating more scenographic shots, one of the many applications that become available is text-to-image storyboard creation. Existing storyboarding tools either extend digital painting applications (e.g. [11]), allow the user to place predetermined objects in a scene to compose the de-

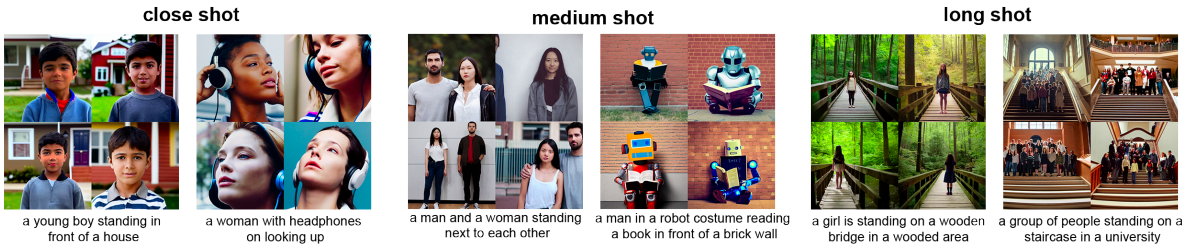
Published in the Proceedings of the Workshops of the EDBT/ICDT 2024 Joint Conference (March 25-28, 2024), Paestum, Italy

*Main author.

† Corresponding author.

tommaso.massaglia@studenti.polito.it (T. Massaglia); bartolomeo.vacchetti@polito.it (B. Vacchetti); tania.cerquitelli@polito.it (T. Cerquitelli)

0000-0001-5583-4692 (B. Vacchetti); 0000-0002-9039-6226 (T. Cerquitelli)



sired frame (e.g. [12]), provide a simple interface to create a reference of the desired scene (e.g. [13]).

For more deep learning-related approaches, StoryGAN [14] generates a sequence of images that describe a story written in a multi-sequence paragraph. To do this, the proposed framework uses a sequential Generative Adversarial Network [15] that consists of a Story Encoder, an RNN-based Context Encoder, an image generator conditioned on the story context, and an image/story discriminator that ensures consistency. Diffusion Models allow for high-quality generation on multiple domains without needing specific training, and a better understanding of the conditional text input than GANs. The conditioning based on previous frames could be a possible approach for increased temporal consistency even in LDMs.

Dynamic Storyboarding [16] approaches the storyboarding task directly by automatically composing scenes out of user inputs by simulating in a virtual environment the scene and discriminating the best proposal out of the available ones. This approach generates rich and complex dynamic (video) storyboards, but it lacks the customizability and intuitiveness that Diffusion Models offer through textual conditioning. Furthermore, by using ControlNet 2.5 trained networks it's possible to add conditioning through more inputs such as scribbles, which at the cost of a slightly higher effort can lead to much better generations.

2.2. Text-to-Image Diffusion Model

Diffusion models are a type of probabilistic generative models that generate samples from a learned distribution by reversing the "diffusion process", modeled as a Markov process of gradual Gaussian noise addition. The generative process is carried by gradually removing noise from a random initial sample. A text-to-image diffusion model ϵ_θ, given an noise map $z_t \sim \mathcal{N}(0,1)$ at timestep t and a conditioning vector $c = \tau_\theta(y)$ generated using text encoder τ_θ and prompt y, generates an image $\epsilon_\theta(z_t, t, \tau_\theta(y))$. During training, the sample generated using the conditioning $\tau_\theta(y)$ is compared to its original counterpart ϵ. The loss is computed as:

$$L_{DM} = \mathbb{E}_{x,\epsilon\sim\mathcal{N}(0,1),t}[||\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))||_2^2], \quad (1)$$

where both τ_θ and ϵ_θ are jointly optimized during training.

2.3. CLIP

CLIP [7], short for *Contrastive Language Image Pretraining*, is a technique developed to approach the zero shot classification task by learning the contents of an image directly from raw text description of it rather than from labels (such as the classes found in the ImageNet dataset). By learning from natural language, the resulting model is much easier to scale compared to standard crowd-sourced dataset thanks to the vast amount of text available on the internet. The representation that is learned with CLIP is tightly connected to language, which enables flexible zero shot transfer. Given a batch of N (text, image) pairs, CLIP is trained to predict which of the $N \times N$ possible pairings across a batch actually occurred. To do this, CLIP learns a multi-modal embedding space by jointly training an image encoder (based on a vision transformer) and a text encoder to maximize the cosine similarity of the image and text embeddings of the N real pairs, while minimizing the cosine similarity of the $N^2 - N$ incorrect pairings.

2.4. Latent Diffusion

Latent Diffusion Models are introduced in [5] which proposes to move the diffusion process from the computationally expensive pixel space to a less intensive latent space. Given an image $x \in \mathbb{R}^{H\times W\times 3}$ in RGB space, the encoder $\mathcal{E}$ encodes x into a latent representation $z = \mathcal{E}(x)$, and the decoder $\mathcal{D}$ reconstructs the image from the latent, giving $\tilde{x} = \mathcal{D}(z) = \mathcal{D}(\mathcal{E}(x))$. Thanks to the latent representation enabled by $\mathcal{E}$ and $\mathcal{D}$, likelihood-based modelling becomes a more suitable task as higher complexity details are abstracted away and the learning can focus on the important semantic bits of the data. Rather than using an autoregressive, attention-based approach, image-specific inductive biases can be taken advantage of. The underlying UNet is built primarily from 2D convolutional layers. Different forms of conditioning can be applied during generation such as image maps and text (which uses CLIP encodings to generate the conditioning tokens); the text-to-image generation process is carried by feeding as input a random noise vector and a textual prompt to the denoising U-net of the model.

2.5. ControlNet

Described in [17], ControlNet is a network structure developed to support additional input conditions in existing diffusion models; rather than controlling the synthesis of images only through text or an input image, ControlNet allows to use of inputs such as canny mapsand depth maps and poses as inputs for the denoising process, even combining them in the same process, allowing for an increased level of control on the output.

ControlNet works by creating a *trainable copy* and a *locked copy* of an existing large diffusion model; the locked copy preserves the network capabilities learned from billion of images, while the trainable copy is trained on task-specific datasets to learn the conditional control. The two networks are then connected using a new type of convolution layer called *zero convolution*. Only the first half of the denoising

U-Net is trained and the encoder blocks are connected to their respective decoder blocks through zero convolutions. Video ControlNet [18] proposes an approach that enhances temporal consistency when converting an existing video using Stable Diffusion.

3. Method

Modern diffusion models can increasingly produce photo-realistic images through conditional generation that are almost indistinguishable from the human eye. The most common form of conditioning is through text (called 'prompt'). By encoding text and using the resulting encodings in the cross-attentional layers of the denoising U-network as conditioning, it is possible to influence the generation process toward a desired outcome. In most cases, however, the amount of control we can exert over the output is limited and requires either specialized prompt engineering or fine-tuning to teach the model how to better represent the desired concept. Extensive fine-tuning can be prohibitively expensive and requires multiple GPU hours on a cluster. To solve this problem, techniques have been developed to quickly add new themes or styles to an existing large diffusion model like DreamBooth[2].

The intuition behind our approach is that learning a shot type is similar in a way to learning a style (if a painter always painted portraits his "style" would always have the subject *close to the camera*), and as such we could use DreamBooth capabilities to teach an existing Latent Diffusion Model what different shot types are.ì

Figure (1) outlines the basic steps we adopted to fine-tune the model. The particular DreamBooth implementation we used leverages Low Rank Adaptation (LoRa) [19] to significantly reduce training time and more easily create shareable checkpoints. The entire process consists of creating a well-constructed dataset, since the quality of the training images and labels greatly affects the output model, selecting a base model for fine-tuning, and creating a ΔW. We refer to the base model as W and the fine-tuned model as W', such that $W' = W + \Delta W$. ΔW contains the learned weights that can then be invoked during inference to be applied to the selected base.

3.1. Training set creation

The training set that is used when finetuning a pre-trained diffusion model is one of the most important contributors to the output quality. As the model learns to reproduce the contents of the training set, by having high-quality samples, the generated image quality will improve as well. Another important aspect of the training set is the caption that is associated with each image. The way DreamBooth adds knowledge to a pre-trained model is by *learning* the concepts of the input image that the original model doesn't already possess in its prior knowledge. In our case, the caption associated with each shot should include a highly accurate description of the shot so that the model would pick up the *concept* of the shot scale and not other already known ones. To reach this goal, which is the creation of a task-specific training set, we define a 5 steps approach that can be applied to any large dataset of movie shots. **(i) Data Collection**: the first step is to acquire a large enough dataset to use as a base; movie shots datasets have a wide range of image quality, so it's suggested to start from a large enough one in order to have a *guarantee* of having enough high-quality samples. **(ii) Filtering**: depending on the metadata available of the chosen dataset, filtering out the lower-quality images, even with arbitrary filters, can largely improve the speed of the subsequent steps. **(iii) Cropping**: the required resolution for images when fine-tuning Stable Diffusion is 1×1, with the most used sizes being 768×768, 512×512 and 256×256. By using a content-aware cropping method it's possible to obtain the necessary image size in a quick way while keeping the most important part of the shot. **(iv) Labeling and shot selection**: as there is no precise enough approach for automatic shot labelling and the shots require close supervision for the quality of the image and the crop, labelling by hand becomes a necessity. By sampling without repetition from the available pool of images and assigning the correct label, it's possible to quickly handpick and label the necessary shots, which should range between 100 and 200 for styles. A good movie variety should be kept to not teach unwanted subjects. **(v) Captioning**: once the required images per shot scale are reached, a first basic caption can be generated by using models such as *blip-2* [20], which also have the advantage of generating captions that resemble the CLIP description style. Once again, human supervision is highly suggested for the generated captions.

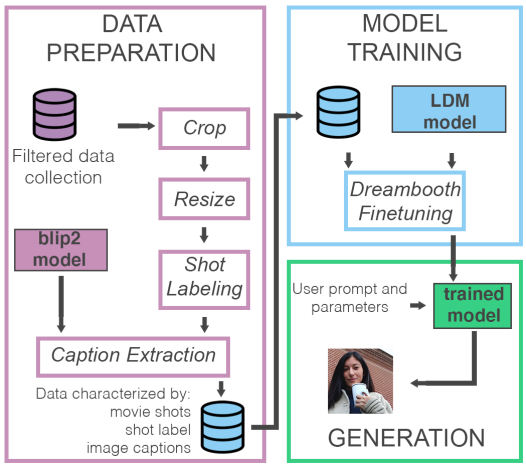


Figure 1: A visualization of the finetuning process using LoRa DreamBooth. To create basic captioning that required minimal human work, Blip2 was used. Labels for shot types were added by hand due to the small number of pictures necessary.

Once the dataset is correctly prepared, the training can begin.

3.2. Model Training

In order to finetune the LDM we used DreamBooth [2]. The idea behind DreamBooth is to, given a few input images ($\approx 3-5$), bind the subject to a *unique identifier* such that when it is used in the prompt along with the class it belongs to (e.g. "A [V] dog"), the prior knowledge of the class is used along the new information to reconstruct the subject. A new autogenous class-specific prior preservation loss is introduced on top of the regular training objective to encourage diversity and counter language drift. During training, the model is supervised *with its own generated samples* in order to retain the prior knowledge of the class and to use it along

with the knowledge of the subject instance to generate new samples.

By itself, DreamBooth already manages to significantly decrease the cost of adding a subject to an existing model. But, as a further optimization, we used Low Rank Adaptation [19] applied to the DreamBooth process [21]. LoRa allows efficient finetuning even in low-power devices while keeping a high-quality end-result. Instead of training the entire model, LoRa works by finetuning the residual: i.e. train ΔW instead of W.

$$W' = W + \Delta W \tag{2}$$

Through matrix decomposition it's possible to further decrease the amount of parameters to finetune, hence reducing the size of the output model by an even larger degree.

$$\Delta W = AB^T \tag{3}$$

The attention layers parameters of the cross-attention layers in the denoising U-Net of Stable Diffusion are enough to tune to obtain the desired output.

Given an existing diffusion model W, a LoRa of it is applied on top in the form of $W' = W + \alpha\Delta W$: when α is 0 the model is the same as the original one when α is 1 the model is the same as the fully finetuned one. Applying this form of optimization to DreamBooth makes it possible to achieve two primary goals: faster and less complex training and a lightweight and more versatile output.

Once the training phase is finished, an output file is produced which contains the weights learned during training. The model is then used alongside the original one that was used as a base during the finetuning process (in this case *stable-diffusion-v1-5*) to synthesize images.

Figure 2: *prompt*: a high-quality close_shot picture of a woman holding a cup of coffee in front of a brick building $\alpha\Delta W$

In our specific case, no unique identifier was specified during training; by not binding the concept to a specific token, the model always generates in the trained style (or shot type in our case) when the ΔW model is specified in the prompt.

The caption in figure (2) is the prompt that was used to generate the picture. The token "$\alpha\Delta W$" is a placeholder control sequence that is added in the prompt to add the weights and layers from the LoRa (ΔW, *closeshot* in this case) to the pretrained full model that's being used for the generation with weight α.

3.3. Generation

Once the model is successfully trained, the generative process can begin. Generation is performed by providing the model with a series of parameters along with a textual prompt describing the scene. The prompt can be either in the positive field, where the generation is *moved towards* the conditioning, or the negative field, where the model generates *away* from the concepts specified in the negative field. Prompt engineering takes a big role in the generative process, with certain prompts such as *"high quality"* and *"masterpiece"* guiding the generated image towards more aesthetically pleasing results. The most meaningful generation parameters are:

- **Sampler**: at each step of the diffusion process a certain amount of noise is predicted and subtracted from the image. The sampler takes care of both computing the predicted noise and scheduling the noise level at each sampling step so that an equally noisy image can be sampled. There are many available with different benefits.
- **Steps**: changes how much noise is subtracted from the image at each step, the larger the number of steps the slower the generation process is, but finer details might be developed this way.
- **CFG Scale**: short for Classifier Free Guidance scale, classifier free guidance is a technique that moves the generated samples away from random unlabeled ones, essentially making the generated image adhere more to the provided prompt.
- **Seed**: determines the initial noise map, different seeds will result in different images.

Furthermore, the value α that determines how much the ΔW model weights are applied takes an important role in the generative process. As there is no deterministically perfect way to train a DreamBooth model, sometimes lowering how much influence the finetune has can improve results.

4. Preliminary Experiments

4.1. Training Set

Among the many available movie repositories, [FILMGRAB][1] was chosen as it provides high quality, hand picked movie frames.

We began by collecting 127.000 shots from 2166 movies. All the pictures with less than 3 color channels were pruned, as well as the ones coming from movies released before 2013 to guarantee a certain degree of image quality and resolution. The shots were then cropped using content-aware image cropping to the size of 512×512 pixels because of computational constraints. Out of the remaining 41.750, only 600 (200 per shot type) were then to be selected. As the number of required pictures is relatively small, shot-type selection and labelling was performed by hand. Randomization was achieved by sampling single shots from all the available ones and by assigning a label, adding it to the training set if and only if the quality and crop were deemed to be appropriate. As the training set is small, the training is very sensitive to bad samples.

[1] Open source for research purposes.

The final step was adding textual captions. To aid in the captioning process, the Vision-Language model *blip2-flan-t5-xl* [20] was used to generate a first *CLIP* [7] *style* caption with human supervision.

4.2. Testing Set

The dataset used for testing is composed of 1800 shots sampled from the filtered 41.750 shots evenly distributed between the three shot types (long shot, medium shot, close shot), and their respective caption generated using BLIP2 [20] without supervision. The generated captions were not supervised for testing purposes. The collected captions were then randomly sampled and used to generate two pictures from the same starting seed N times, one with and one without training, for a total of 1500 pairs of "trained" and "non-trained" images, evenly split between shot types, with generation parameters 2.

Table 2
The pararameters used for generation during testing

sampler	DPM++ SDE Karras
steps	16
seed	random
cfg_scale	6
prompt	a high-quality [shot_type] picture of [caption]
size	512 x 512

4.3. Metrics

To get a quantitative result two metrics were adopted following in the footsteps of the original DreamBooth [2] implementation. The first one is CLIP-T [7], the average pairwise cosine similarity between the clip embeddings of the generated image and the prompt that generated it. The second metric, DINO [8], measures the average pairwise cosine similarity between the ViTS/16 DINO embeddings of generated and real images, essentially measuring how similar the generated image is to its real counterpart. The results shown in 3 show a slight (although significant for the considered metrics) increase for both the CLIP-T and DINO scores over the baseline model. The lower increase seen in the CLIP-T compared to the DINO metric is justified as the model doesn't learn to represent more concepts (so from a *CLIP perspective* the objects present in the picture are *the same*) with our finetuning, but instead learns to represent them closer to the training image, especially from a camera distance perspective. From a qualitative analysis, it appears that the fine-tuned model is more often able to generate images that are semantically close to the prompt used to generate them. Sometimes it even generates elements that are present in the prompt that the baseline model ignored (e.g., a person when two were specified, a car that is not present). In addition, since there is *no free lunch*, although it has not been tested on other tasks, we expect the fine-tuned model to perform worse on other generative tasks, and in the generated examples we can see that it more often generates faces similar to those shown during training.

As a secondary and ablation study, 600 additional image pairs were generated using the same setup as before, but removing all information regarding the acquisition type from the text conditioning. Looking at the results of the DINO score in Table 4, it can be seen that the images generated with the fine-tuned model still have a higher DINO score than the baseline, indicating that the model generates images at the specific fine-tuning scale even without guidance.

	CLIP-T	DINO
baseline	0.3221	0.4163
ours	**0.3269**	**0.4989**

Table 3
Results for the CLIP-T and DINO metrics on the 1500 pairs test.

	CLIP-T	DINO
baseline	0.3214	0.4014
ours	**0.3234**	**0.4803**

Table 4
Results for the CLIP-T and DINO metrics on the ablation test.

4.4. Qualitative Survey

We conducted in addition a survey of human subjects. Each subject was shown a total of 36 pairs of images A and B generated with the same setting and prompt, one from the baseline model and one from the finetuned one. Whether an image was labelled A or B was randomized. The generated patterns were monitored in a very light form to ensure that the images were safe for all. Each image pair was shown along with its associated shot type and generator prompt. For each image pair, three questions were asked: (i) Which image do you like best?; (ii) Which image corresponds more to the associated shot type?; (iii) Which image corresponds more to the associated prompt?

The possible answers for each question were A, B, or *neither/same* if the two images were considered equivalent in some aspect. A total of 55 subjects responded to the survey, and the results are reported in Table 5. It can be seen that even with human evaluation, our approach generates images that are more appealing and closer to the associated shot type and prompt in almost or more than half of the cases.

Table 5
Results collected from a survey conducted on 52 subjects. The score are expressed as percentage over the total number of answers.

question	baseline	ours	same / neither
Which picture do you like most?	26.18	**57.43**	16.4
Which picture is closer to the associated shot type?	20.46	**56.84**	22.7
Which picture is closer to the associated prompt?	20.35	**49.31**	30.34

Aside from image likability, the baseline model obtained the lowest score of the three, indicating that the generation is of equal quality to the generation without fine-tuning in most cases. The results are consistent, comparing the survey to CLIP -T and DINO metrics. The higher likeability and shot-type closeness are directly related to DINO and are noticeably higher than prompt closeness and CLIP-T compared to the baseline.

Figure 3: Some examples of the generation of the same subject with the three different trainings (close, medium, and long shot) with different levels of α

5. Conclusions and Future Developments

We have presented an approach that uses novel techniques such as DreamBooth and LoRa to finetune an existing latent diffusion model to generate specific types of shot types. Based on the intuition that learning a shot type is similar to learning a style, which DreamBooth was shown to be capable of, we achieve improvements in both compliance and similarity of reference images by using only 200 images for each shot type, as shown by CLIP -T, DINO, and even human evaluation metrics. We test our approach on a storyboarding task showing the potential uses of modern LDMs in video production, mainly when supported by domain-specific training. Furthermore, novel techniques, such as ControlNet open the doors to even more specific conditioning forms. Developments such as [18] show the power that ControlNet offers, and applying the technique for cinematic purposes could be an interesting development point. Regarding our work, as DreamBooth training is far from a solved task, more tests could yield even better results.

References

[1] R. M. Stability AI, Stable diffusion release blog post, https://stability.ai/blog/stable-diffusion-public-release, 2022. (accessed 23-May-2023).

[2] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, K. Aberman, Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation, 2023. arXiv:2208.12242.

[3] R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, D. Cohen-Or, An image is worth one word: Personalizing text-to-image generation using textual inversion, 2022. arXiv:2208.01618.

[4] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, 2020. arXiv:2006.11239.

[5] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, 2022. arXiv:2112.10752.

[6] B. Rooney, K. E. Bálint, Watching more closely: Shot scale affects film viewers' theory of mind tendency but not ability, Frontiers in Psychology 8 (2018). doi:10.3389/fpsyg.2017.02349.

[7] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, I. Sutskever, Learning transferable visual models from natural language supervision, 2021. arXiv:2103.00020.

[8] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, A. Joulin, Emerging properties in self-supervised vision transformers, 2021. arXiv:2104.14294.

[9] E. Molad, E. Horwitz, D. Valevski, A. R. Acha, Y. Matias, Y. Pritch, Y. Leviathan, Y. Hoshen, Dreamix: Video diffusion models are general video editors, arXiv preprint arXiv:2302.01329 (2023).

[10] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni, D. Parikh, S. Gupta, Y. Taigman, Make-a-video: Text-to-video generation without text-video data, ArXiv abs/2209.14792 (2022).

[11] Storyboarder, https://wonderunit.com/storyboarder/, ????.

[12] Storyboardthat, https://www.storyboardthat.com/, ????.

[13] Studiobinder, https://www.studiobinder.com/storyboard-creator/, ????

[14] Y. Li, Z. Gan, Y. Shen, J. Liu, Y. Cheng, Y. Wu, L. Carin, D. Carlson, J. Gao, Storygan: A sequential conditional gan for story visualization, 2019. arXiv:1812.02784.

[15] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, 2014. arXiv:1406.2661.

[16] A. Rao, X. Jiang, Y. Guo, L. Xu, L. Yang, L. Jin, D. Lin, B. Dai, Dynamic storyboard generation in an engine-based virtual environment for video production, ArXiv abs/2301.12688 (2023).

[17] L. Zhang, M. Agrawala, Adding conditional control to text-to-image diffusion models, 2023. arXiv:2302.05543.

[18] E. Chu, S.-Y. Lin, J.-C. Chen, Video controlnet: Towards temporally consistent synthetic-to-real video translation using conditional image diffusion models, 2023. arXiv:2305.19193.

[19] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, 2021. arXiv:2106.09685.

[20] J. Li, D. Li, S. Savarese, S. Hoi, Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. arXiv:2301.12597.

[21] S. R. aka cloneofsimo, lora, https://github.com/cloneofsimo/lora, 2023.