Adapting Text-to-Image Models for Video Generation Using Few Shot Learning

Samir Agarwala **Hong Ju Jeon** **Jared Watrous**
Stanford University
{samirag, hjeon, jwat2002}@stanford.edu

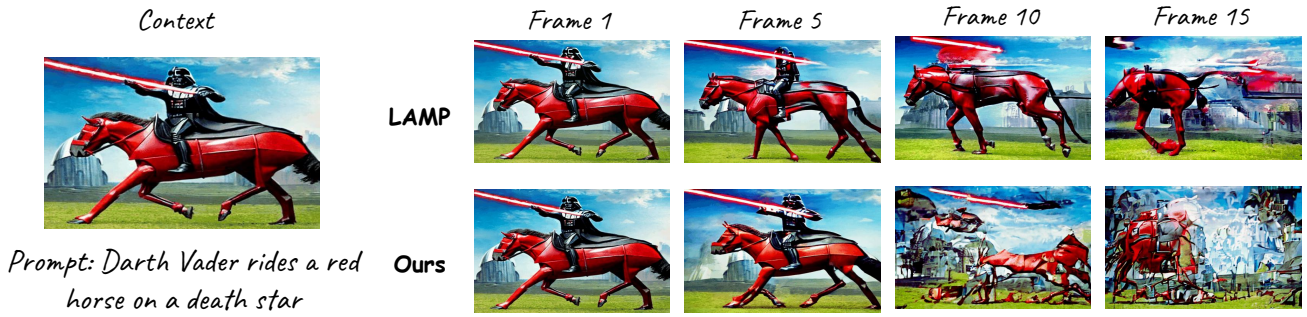

Figure 1: Video generative models can be used to generate vivid videos of arbitrary text prompts. Here we show videos generated by LAMP [18], our closest baseline, and our method that has been finetuned with an auxillary mask loss wherein we predict masks from decoded latents during training. Both methods find it challenging to generate consistent motion in videos and eventually degenerate, but we do see that LAMP has a more consistent video with more realistic motion. More work needs to be done in this area so that we can develop methods that can efficiently finetune existing text-to-image models to generate videos for arbitary prompts.

1 Introduction

We live in a 4D world where objects are in different states of motion and are taking various actions to interact with the environment around them. However, the actions and motion which different actors take are not arbitrary. Instead, they follow some general norms which we would expect to see. For instance, if we saw a video of an horse running without its leg moving on land, we might look in disbelief since that is simply not possible. However, if we saw a horse running with its legs in motion, we would consider that typical behavior depicted by a horse. Thus, there are specific kinds of actions and movements we expect to see from different actors. In this work, we tackle the challenging problem of text-conditioned video generation where we are given a text prompt of a certain category of motion and want to be able to generate a realistic video that aligns with the given prompt.

In the past few years, the advent of diffusion models [15] has allowed us to generate high-quality images of various different things. These models are trained on internet-scale image datasets and learn to model the distribution of images, through which they are able to synthesize novel images at inference time. However, generative models for video generation still have a long way to go. Compared to image datasets, video datasets are more challenging to collect and require significantly more compute to train large models on which can make the task unscalable on internet sized datasets. Additionally, videos have more complex structure than images since they need to be coherent and have temporal consistency. The larger computational and data costs of modelling videos along with the inherently more challenging nature of the problem compared to image generation makes the problem challenging and there is a lot of scope in pursuing work in efficiently learning generative models for videos.

Code is available at `https://github.com/samiragarwala/VideoGeneration`

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

Recent work in video generation has focused on adapting text-to-image (T2I) models for text-to-video (T2V) generation [7, 17, 18, 11]. However, we see that zero-shot approaches [11] suffer from temporal inconsistency. Other approaches finetune text-to-image models on one- [17] or a few videos [18] of a motion category in the hope of learning a motion prior. These methods have much better temporal consistency than zero-shot approaches. There is still work to be done in this area as we often observe issues related to foreground and background motion being inconsistent across frames.

In this work, we build on progress in finetuning text-to-image images in the few-shot domain by encouraging our generative model to disentangle foreground and background motion in the hope of being able to learn more temporally consistent motion priors as a result of the disentanglement. We focus on few-shot finetuning of T2I models since that is not only scalable but also seems like a promising direction to explored based on previous work [17, 18]. Since, previous work suffers from foreground and background motion inconsistency in generated content, we explore ways to mitigate this issue by incorporating off-the-shelf mask supervision as an auxiliary loss while finetuning T2I models for T2V. Our key insight is that *encouraging the generative model to reason about foreground masks during training may allow it to learn a more disentangled representation of foreground and background motion*. Thus, in this work, we examine how adding mask supervision as an auxiliary loss during finetuning of T2I models for video generation can impact output quality.

2 Related Work

Our work builds on recent progress in text-to-image diffusion models and video generative models.

Text-to-image diffusion models. Diffusion models [5] have had tremendous success in image generation tasks and been able to perform better than previous models such as generative adversarial networks and variational autoencoders. One area of particular interest in image generation has been that of text-to-image generation where the generative models is given a prompt in natural language and expect to generate an image aligned with the prompt. Diffusion models [15, 14] have had huge success in this area too. Stable Diffusion [15] has approached this task by learning a latent representation of images through a variational autoencoder and then learning a generative model over the learned latent space for efficiency, while DALLE-2 [14] learns a diffusion prior in the CLIP [13] encoding space and then decodes those images to synthesize images. Although these models have been able to generate both vivid and abstract images based on user prompts, it is non-trivial to extend them to generate videos without training on video data.

Video generative models. In the past few years, there has been progress in developing generative models for modelling videos. Most of these efforts have approached the problem using autoregressive model [10, 1] where video frames are generated sequentially while other efforts have combined latent variables for efficiency with autoregressive priors [12] to get temporally consistent generation with tractable likelihood computation. However, these models are often trained on limited datasets and are unable to capture diverse video settings.

Since diffusion models have been proposed, there has been work using them in video generation [4, 6, 2]. For instance, [6] expanded image diffusion models to deal with videos while [2] approaches the long-horizon unconditional video generation task. Recent work has focused on diffusion models for T2V generation [4, 17, 16, 18, 11, 7]. Although initial works trained or finetuned directly on video datasets [4, 16], in the past year, we have seen a growing effort to directly leverage pretrained T2I models such as Stable Diffusion [15] for T2V generation in a zero-shot manner by performing latent space warping [11] or in the few-shot domain by finetuning a few weights of T2I models on one or more videos to learn motion priors [17, 18]. The most promosing of these methods is LAMP [18] which adds spatio-temporal attention layers to T2I models to allow video generation, and then finetunes them on a few videos to learn category level motion priors. The generation from LAMP are quite realistic, however as the authors note the foreground motion often impacts the consistency of the background. In our work, we propose using off-the-shelf segmentation masks as supervision to try and encourage our generative model to disentangle foreground and background motion.

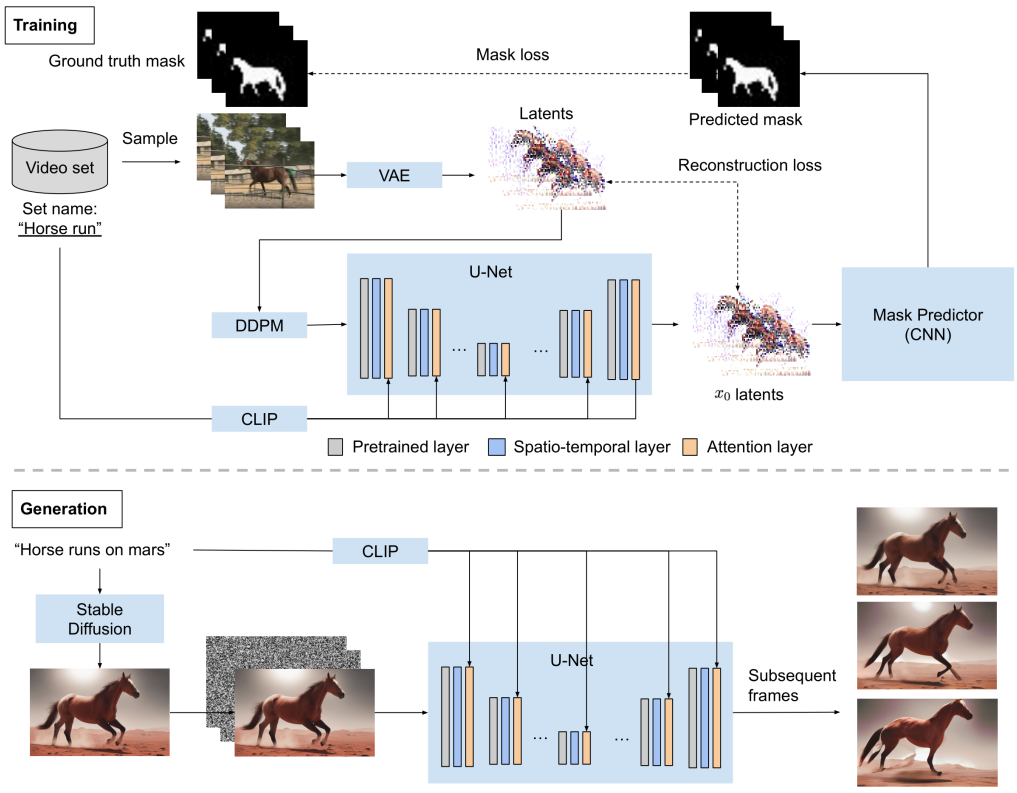

Figure 2: Architecture diagram of our modified LAMP [18] model. We add a Mask Predictor network that takes in the denoised outputs from the U-Net and predicts a segmentation mask of the subject in the image. During training, we exclude the first frame when calculating the losses. At inference time, we generate the first frame of the video using Stable Diffusion [15] and perform the reverse diffusion process to generate video frames from sampled noise.

3 Approach

We build on the few-shot T2V generation method LAMP [18] by including mask supervision to help the model learn to disentangle foreground and background motion in the hope of generating more consistent videos. We first discuss LAMP, the method we build on, before explaining how we propose to incorporate mask supervision into our approach.

3.1 LAMP Overview

LAMP [18] adapts Stable Diffusion [15] for video generation by finetuning the T2I model on 8 to 16 videos of a particular motion category. In this approach, the first frame of the video is generated from a T2I model and used to guide the video generation process. The main components of the approach include inflating weight sizes to allow processing of videos, the inclusion of a novel spatio-temporal layer and modifying attention layers so that all frames attend to the first frame to ensure consistency of content. We refer readers to the paper for any additional details.

Spatio-temporal layers. LAMP [18] adds a novel feature layer to learn the motion dynamics in both the temporal and the spatial dimensions. To learn temporal features that are useful in next frame prediction, 1D convolution is used with a kernel that slides over the frames dimension $\{f^{i-1}, f^i, f^{i+1}\}$ for all i in the latent vector of shape (b, f, c, h, w). To capture the spatial features, a 2D convolution is used on the latent vector of shape $(b * f, c, h, w)$ with an output channel of 1. The two features are then concatenated and learned through fine-tuning of the video dataset.

Attention layers. LAMP modifies the attention layers such that all generated frames attend to the first generated frame.

$$Attention(Q^i, K^1, V^1) = \frac{Softmax(Q^i(K^1)^T)}{\sqrt{(d)}}V^1$$

This modification has been shown in prior works to establish consistency in generated frames [11]. Moreover, there is an additonal self-attention layer on the temporal dimension. The two outputs are added together to produce the attention score.

3.2 Adding Mask Supervision

We add a Mask Prediction Network on top of [18] that predicts a segmentation mask that separates the subject and the background. As the mask is learned through supervision, the gradients are backpropagated through the U-Net. Segmentation masks have been proposed to improve consistency over the generated frames in a postprocessing steps [11], but to the best of our knowledge, no approaches have incorporated using the mask when learning the features in the spatio-temporal layer of the denoising U-Net.

Data Preparation. To prepare the labeled training data containing image frames and segmentation mask pairs, we iterate through each frame of the training videos and utilize an off-the-shelf OneFormer [9] segmentation model to create a sequence of segmentation masks corresponding to the video. We use bilinear interpolation to downsize the predicted size to match the shape of the latents.

Mask Prediction Network. We add a Mask Prediction Network on the existing LAMP [18] architecture that outputs a binary segmentation mask of the subject in the image. This network is a series of three 2D-convolution layers that preserve the input image's width and height and operates independently on each frame.

During training, we input the approximate denoised latent vector $\hat{x}_0$ into our proposed mask predictor. This approximation is computed by using equation 15 from Ho et. al [5]:

$$x_0 \approx \hat{x}_0 = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_0(x_t)}{\sqrt{(\bar{\alpha}_t)}}$$

where $\epsilon_0(x_t)$ represents the predicted noise added to x_{t-1} in the noising process and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. At inference time, we iteratively denoise latents as in LAMP to generate the output video and predict masks using the x_0 latent which we obtain by performing T denoising steps on the noisy latent x_T.

Model Training. To train the network we have a combination of two losses:

$$L = \sum_{f \in \textit{frames}} L^f_{rec} + \alpha L^f_{mask}$$

where $L^f_{rec} = \|\epsilon^{\mathbf{f}} - \hat{\epsilon}^{\mathbf{f}}\|_2$ is the mean squared error of the sampled and predicted noise for frame f following [18]. We also include $L^f_{mask} = -(y\log(p) + (1-y)\log(1-p))$, which is the pixel-wise binary cross entropy between the ground-truth and predicted mask for each frame to encourage foreground and background disentanglement, with $y \in \{0, 1\}$, and p being the predicted probability of the pixel representing foreground. During training, we sum over the loss on all frames except the first frame which is not a model prediction and instead is the output of a T2I model that is used to guide generation.

Video Generation. During inference, we remove the Mask Predictor Network and only use the finetuned U-Net. We take the user query to create the first frame's image using a pre-trained T2I model and also the CLIP embedding that is used to guide the image generation. The subsequent frames are denoised together while being conditioned on the first frame and CLIP embedding to create a video sequence that aligns with the given prompt.

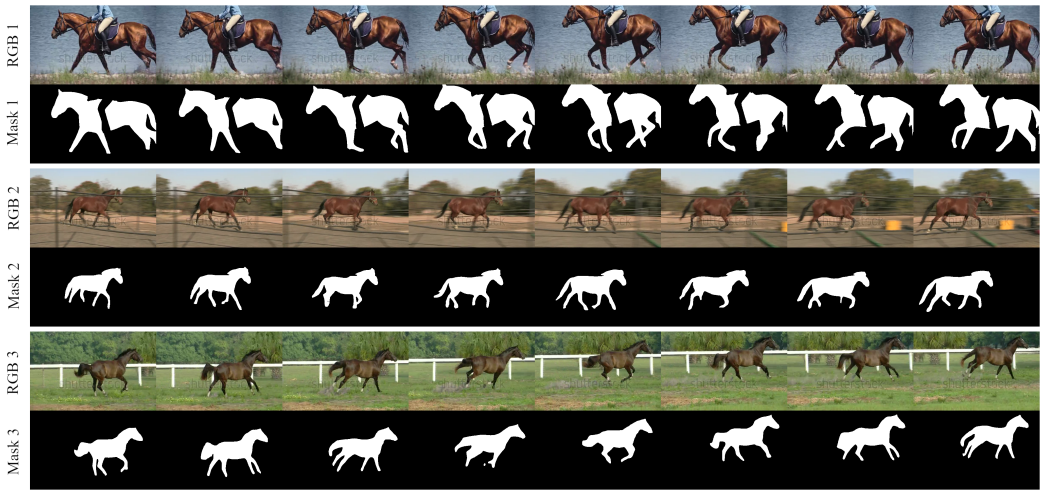

Figure 3: Sample frame-mask pairs from our dataset. The original RGB videos are taken from the "horse run" motion class of the LAMP dataset [18]. We obtain segmentation masks by running OneFormer [9] on each frame.

4 Results

Dataset. Since we are building on an existing T2I model to learn category-level motion, we do not require much training data. We use the dataset from LAMP [18], specifically the "horse run" class of motion, containing 8 RGB videos. We augment this dataset for our use case by generating segmentation masks for each frame using OneFormer [8]. See Figure 3 for a few samples of these RGB-mask pairs. Although the LAMP dataset contains 64 videos with 8 total classes of motion (e.g. "horse run," "birds fly," "firework," etc.), we do not train or evaluate our work on all categories due to the limited availability of computation resources.

Baselines. We consider two existing T2V models as baselines. Our first baseline is Text2Video-Zero (T2V-Zero) [11], a zero-shot adaptation of a T2I model for video generation. T2V-Zero generates the first frame using the typical T2I pipeline and stores the latent vector x_0^1. To construct the kth frame, it then applies a warping function W_k to the original latent x_0^1, yielding $x_0^k = W_k(x_0^1)$. The warping function is a simple translation by a vector $\delta^k = \lambda \cdot (k-1)\delta$, where λ is a hyperparameter controlling the global rate of the motion. Notably, T2V-Zero does not train on any video data; the fidelity of the motion relies entirely on the structure of the latent space and assumes a linear translation of the latent produces a reasonable sequence of frames.

Our second baseline model is the pretrained LAMP [18] model, a few-shot T2V model that makes no assumptions about the structure of the latent space. LAMP augments the model's U-Net by inserting spatio-temporal and attention layers that are fine-tuned on a finite set of reference videos, using the video frames as the batch. In our case, we train on the 8 original "horse run" videos described in the previous section. Once trained, LAMP can generate videos by constructing the first frame using a typical T2I image model and conditioning the predicted noise for all other frames on this context image. Our model is built on top of LAMP by adding the mask predictor head, with the intent of disentangling foreground and background motion in the spatio-temporal and attention layers.

In addition, we provide our own baseline model (*Ours-Baseline*) where we condition LAMP [18] on both a segmentation mask and the context image. For this, we concatenate the context latent vector with the mask and run it through a 2D convolution (with kernel size 1) which maps the 5-channel input to the 4-channel latent vector expected by the vanilla LAMP model. Although not an optimal baseline, this allows us to see how we may be able to incorporate information from a segmentation mask as a conditioning input while finetuning existing T2I models for video generation.

Evaluation. We follow the quantitative evaluation criteria proposed by previous T2V methods [18, 11] to measure text-video alignment, frame consistency, and generative diversity. Similar to

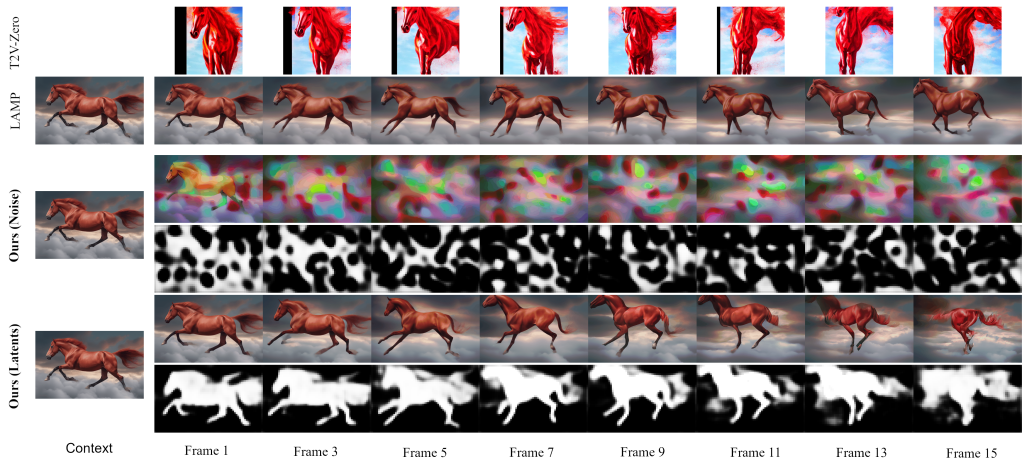

Figure 4: Frames generated with the prompt "a red horse runs in the sky" using Text2Video-Zero [11], LAMP [18], and our method, as well as the predicted masks generated by our method (last row). In LAMP and our method, Frame 0 is generated by a traditional T2I model and used as an input to our model.

LAMP, we test the text-video alignment by computing the mean CLIP score [3] computed across all generated frames. To measure consistency, we follow [17]'s approach and measure the mean cosine similarity between CLIP embeddings of every possible frame pair in a video, then average this score across all videos. Lastly, to measure overall generative diversity, we follow [18] and represent a video by the mean CLIP embedding of its frames, then compute the mean cosine similarity across all possible video pairs. In this case, a lower score indicates higher diversity, as the metric effectively measures the similarity between videos. Notably, whereas LAMP computes the above metrics across all 8 classes of motion, we only compute it across the 6 provided prompts of the "horse run" class from LAMP's evaluation set, due to the computational cost of training individual models for each class.

Implementation Details. We use a T4 GPU with 16GB VRAM to train all three experiment. We trained the Mask Predictor on x_0 latents and the mask-conditioned baseline for 10,000 iterations while the Mask Predictor that took noise as an input was trained for 6,000 iterations. Our base pretrained weights are loaded from Stable Diffusion v1.4. For the supervised mask loss weight, we set loss weight $\alpha = 1$ in our loss function. The learning rate for the Mask Prediction Network and the U-Net are both 3×10^{-5}. We refer readers to LAMP [18] for other parameters.

4.1 Ablation Study on Mask Predictor Inputs

In our preliminary experiments, we used the noise from the U-Net directly as an input to the Mask Prediction Network 3.2 as we believed that the predicted noise may contain sufficient structure to disambiguate foreground and background. However, we found that the noise from the U-Net by itself did not contain enough meaningful spatial information for our network to predict a segmentation mask. This is observed qualitatively in Figure 4, where the generated videos produced segmentation masks that looked more like pure noise than a horse. This was reflected in the generated frames where we briefly saw a semblance of a horse for the first few frames, but then eventually the rest of the frames devolved into correlated noise. Meanwhile, there were significant improvements to the generated videos when we used x_0 as the input to the mask predictor. We observed that the segmentation masks were accurately generated for the first few frames and also the generated frames consistently showed the original horse up until the last few frames.

These qualitative observations are also reflected quantitatively in Table 1, where all metrics showed improvements when using x_0 as the input. This shows that the there are meaningful spatial information contained in the latent vector x_0 that is helpful for predicting a segmentation mask.

Model	Alignment ↑	Consistency ↑	Diversity ↓
Mask Predictor-Noise	23.3654	94.9083	96.6569
Mask Predictor-x_0	**29.2799**	**97.7881**	**86.6374**

Table 1: Ablation study comparing impact of directly using the noise predicted from the model instead of the denoised latents as the input to the proposed Mask Predictor. Directly using noise as an input to the mask predictor is unable to adequately learn motion priors.

4.2 Qualitative Results

As we can observe from Figure 4, our model is able to generate a video of a horse running that is mostly consistent with the given context frame. The motion shown in the video shows the horse extending and protruding its legs in a galloping motion that we observed in the training data. The first half of the generated video frames maintain the same texture and fidelity as the original context image, but in the latter half of the generated frames, the quality of both the foreground and the background decreased. In the latter half, the textures for both the foreground and background became more abstract for all 6 benchmark prompts that we tested, making it seem like the horse is missing body parts or fading away. Furthermore, we observe that the segmentation mask produced by our model was able to accurately separate the foreground and the background. However, the mask also dissolved into an abstract shape towards the end of the video in a correlated way to the generated video frames.

In Figure 5, it is interesting to note that in LAMP, the prompt "Guan Yu rides a red horse" produced a video that showed one horse dissolving into two horses, but in our model, the generated video on the same benchmark consistently showed only one horse through the duration of the video.

Figure 5: Frames generated with the prompt “Guan Yu rides a red horse” using LAMP [18], and our method (using latents).

4.3 Quantitative Results

Table 2 compares the quantitative results of our model against those of Text2Video-Zero, LAMP and our mask conditioned baseline. Notably, the metrics suggest that our model is competitive against LAMP. We outperform LAMP by a small amount in the alignment metrics while LAMP is more competitive in the consistency and diversity metrics. Text2Video-Zero has better diversity metrics than all other methods, but this comes at the cost of consistency, as videos from the method do not take motion priors into account. Our mask-conditioned baseline generally performs worse than other methods across all metrics, which reflects the extremely noisy outputs we observed.

Model	Alignment ↑	Consistency ↑	Diversity ↓
Text2Video-Zero	27.8419	94.1048	**82.2559**
LAMP	29.0095	**98.1169**	86.5104
Ours-Baseline	21.7606	96.5034	99.8242
Ours-x_0	**29.2799**	97.7881	86.6374

Table 2: Evaluation metrics for Text2Video-Zero [11], LAMP [18], our mask conditioned baseline (*Ours-Baseline*) and our proposed mask predictor network on x_0 (*Ours-*x_0). Note that the original LAMP paper [18] computes diversity metrics across 8 classes of motion, while we only compute them across “horse run,” hence our higher diversity metrics are expected to be worse.

4.4 Analysis and Next Steps

From our results in sections 4.1, 4.2 and 4.3, we observe the following:

1. The proposed mask prediction network does not generate coherent output when using the noise from the diffusion model directly as its input but learns to predicts correct masks when using the denoised latents x_0 as the inputs
2. From the qualitative analysis, it seems that our proposed method on x_0 is able to learn to predict foreground masks which generalises to unseen test samples as shown in Figure 4
3. Our proposed method on x_0 is competitive with LAMP on metrics but we do observe that LAMP seems to perform better qualitatively

The results show us that our proposed mask predictor is able to learn the notion of foreground and background through the inference results of the mask predictor but those results do not seem to be helping our model to learn better motion priors. We believe that this result may arise for a few reasons including the design of our loss function, our network architecture choices, and our training length.

The first potential issue with our method may have been that we added an auxiliary mask loss to our loss function with an equal weight as the vanilla reconstruction loss used by diffusion models. The equal weighting of these losses may have resulted in our model not giving sufficient importance to the noise component of the loss and instead paying more importance to learning a good notion of a segmentation mask. Although, we did not have time and compute resources available to investigate this issue, we believe that future work in reducing the importance of the mask loss in the loss function may lead to more promising results.

The second issue that may have led to our method not giving expected results may have been our architecture design. We started investigating the impact of adding an auxiliary mask loss to LAMP [18] by incorporating a simple convolutional mask predictor on the vanilla LAMP Model. Although this is a simple design choice and easier to investigate, it can lead to issues such as not learning to model the motion distribution. Thus, we believe that different design choices may allow future work to more effectively incorporate foreground and background masks during training. For instance, one possibility may be to divide the spatio-temporal motion learning layers from LAMP into a foreground and background component and learn those through off-the-shelf mask supervision. The key difference between what we did and the proposed method would be that here we are explicitly encouraging the model to learn a disentangled representation of foreground and background motion, while in our work we operated on the network outputs which may not have had the intended impact as seen in our results.

The third potential issue may have been our training length for our proposed method. During our project, we were limited by the amount of GPU compute available and the limited availability of credits. Consequently, we trained our models for 6,000 or 10,000 iterations, and compared against the pretrained LAMP model which was trained for 30,000 iterations. Due to this, our results may not have been the most fair comparison, but the best we could do given the limited compute available to us. For future work, we would also recommend training our models for longer to see if the presented metrics improve since it may be possible that after adding a mask predictor, there is a need for longer training to learn good motion priors.

5 Conclusion

In this work, we proposed a way to incorporate auxillary mask supervision while finetuning text-to-image models for few-shot text-to-video generation. Although our method did not outperform our baseline method LAMP [18], we do want to note that this area remains an exciting direction for future work and it is promising that we were able to predict segmentation masks from denoised latents in our results. Training video generative problems is a challenging problem and it remains important to see how we can scalably train models for this by leveraging existing work such as text-to-image models which are ubiquitous today. We encourage future work to build on the areas related to loss function design, network architecture, and training length that are discussed in section 4.4 and continue examining the potential of using off-the-shelf supervision such as masks or optical flow in learning motion priors in the few-shot domain.

References

[1] Mohammad Babaeizadeh, Mohammad Taghi Saffar, Suraj Nair, Sergey Levine, Chelsea Finn, and Dumitru Erhan. Fitvid: Overfitting in pixel-level video prediction. *arXiv preprint arXiv:2106.13195*, 2021.

[2] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. *Advances in Neural Information Processing Systems*, 35:27953–27965, 2022.

[3] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning, 2022.

[4] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models, 2022.

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020.

[6] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models, 2022.

[7] Hanzhuo Huang, Yufan Feng, Cheng Shi, Lan Xu, Jingyi Yu, and Sibei Yang. Free-bloom: Zero-shot text-to-video generator with llm director and ldm animator, 2023.

[8] Jitesh Jain, Jiachen Li, MangTik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. Oneformer: One transformer to rule universal image segmentation, 2022.

[9] Jitesh Jain, Jiachen Li, MangTik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. OneFormer: One Transformer to Rule Universal Image Segmentation. 2023.

[10] Nal Kalchbrenner, Aäron Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Video pixel networks. In *International Conference on Machine Learning*, pages 1771–1779. PMLR, 2017.

[11] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators, 2023.

[12] Manoj Kumar, Mohammad Babaeizadeh, Dumitru Erhan, Chelsea Finn, Sergey Levine, Laurent Dinh, and Durk Kingma. Videoflow: A conditional flow-based model for stochastic video generation. *arXiv preprint arXiv:1903.01434*, 2019.

[13] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[14] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[15] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[16] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data, 2022.

[17] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation, 2023.

[18] Ruiqi Wu, Liangyu Chen, Tong Yang, Chunle Guo, Chongyi Li, and Xiangyu Zhang. Lamp: Learn a motion pattern for few-shot-based video generation, 2023.