

# TRAINING-FREE GUIDANCE IN TEXT-TO-VIDEO GENERATION VIA MULTIMODAL PLANNING AND STRUCTURED NOISE INITIALIZATION

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

Recent advancements in text-to-video (T2V) diffusion models have significantly enhanced the visual quality of the generated videos. However, even recent T2V models find it challenging to follow text descriptions accurately, especially when the prompt requires accurate control of spatial layouts or object trajectories. A recent line of research uses layout guidance for T2V models that require fine-tuning or iterative manipulation of the attention map during inference time. This significantly increases the memory requirement, making it difficult to adopt a large T2V model as a backbone. To address this, we introduce **VIDEO-MSG**, a training-free Guidance method for T2V generation based on Multimodal planning and Structured noise initialization. VIDEO-MSG consists of three steps, where in the first two steps, VIDEO-MSG creates VIDEO SKETCH, a fine-grained spatio-temporal plan for the final video, specifying background, foreground, and object trajectories, in the form of draft video frames. In the last step, VIDEO-MSG guides a downstream T2V diffusion model with VIDEO SKETCH through noise inversion and denoising. Notably, VIDEO-MSG does not need fine-tuning or attention manipulation with additional memory during inference time, making it easier to adopt large T2V models. VIDEO-MSG demonstrates its effectiveness in enhancing text alignment with multiple T2V backbones (VideoCrafter2 and CogVideoX-5B) on popular T2V generation benchmarks (T2VCompBench and VBench). We provide comprehensive ablation studies about noise inversion ratio, different background generators, background object detection, and foreground object segmentation.<sup>1</sup>

## 1 INTRODUCTION

Recent advances in text-to-video (T2V) diffusion models (Yang et al., 2024b; Team, 2025; Kong et al., 2024; Runway, 2024; Agarwal et al., 2025; Bruce et al., 2024; Team, 2024b; Parker-Holder et al., 2024; Inc., 2025; Polyak et al., 2025; OpenAI, 2024) have dramatically improved the quality of generated videos in diverse domains. However, even recent T2V generation models still often struggle to follow text descriptions accurately, especially when the prompt requires accurate control of spatial layouts or object trajectories. As illustrated in fig. 1 (b), recent work has studied improving text alignment by providing detailed layout guidance as an additional input to T2V models, such as bounding boxes (Lin et al., 2024; Long et al., 2024; Lian et al., 2023), optical flow (Liao et al., 2025), and object trajectories (Zhang et al., 2024b; Wu et al., 2024a), which are often created from a large language model (LLM). However, since the original T2V models do not understand the layout guidance, these approaches fine-tune the T2V models with layout annotations (Lin et al., 2024; Long et al., 2024) or iteratively manipulating the attention map of T2V models during inference time (Lian et al., 2023). While effective, these techniques substantially increase memory consumption at inference time or require retraining for different T2V backbones, limiting their scalability to large T2V models.

To address this, we introduce **VIDEO-MSG**, **M**ultimodal **S**ket**S**h Guidance for video generation, a training-free guidance method for T2V generation based on multimodal planning and structured

<sup>1</sup>We provide code in the supplementary materials.

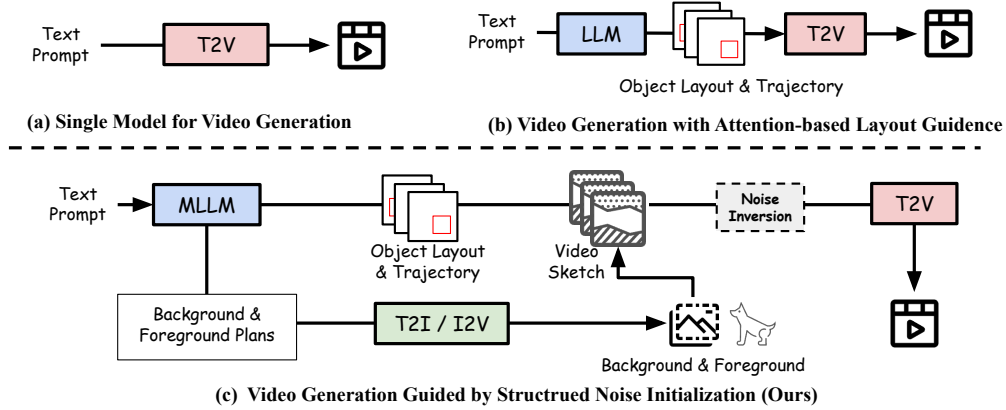


Figure 1: Comparison of different text-to-video generation methods: (a) **single model for video generation**, (b) **video generation with (attention-based) layout guidance**, and our (c) **VIDEO-MSG**, a training-free guidance method for T2V generation based on multimodal planning and structured noise initialization. Since VIDEO-MSG does not need fine-tuning or additional memory during inference time, it is easier to adopt large T2V models than previous video layout guidance methods based on fine-tuning or iterative attention manipulation.

noise initialization, as illustrated in fig. 1 (c). As illustrated in fig. 2, VIDEO-MSG consists of three steps: (1) background planning (section 3.1), (2) foreground object layout and trajectory planning (section 3.2), and (3) video generation with structured noise inversion (section 3.3). From the first two steps, VIDEO-MSG creates VIDEO SKETCH, a fine-grained spatial and temporal plan with a set of multimodal models, including multimodal LLM (MLLM), object detection, and instance segmentation models. Then in the last step, VIDEO-MSG guides a downstream T2V diffusion model with VIDEO SKETCH through structured noise inversion and denoising. Notably, VIDEO-MSG does not need fine-tuning or additional memory during inference time, making it easier to adopt large T2V models, compared to existing methods based on fine-tuning or iterative attention manipulation.

VIDEO-MSG demonstrates their effectiveness in enhancing text alignment with multiple T2V backbones (VideoCrafter2 (Chen et al., 2024) and CogVideoX-5B (Yang et al., 2024b)) on popular T2V generation benchmarks (T2VCompBench (Sun et al., 2024) and VBench (Huang et al., 2024)). For example, VIDEO-MSG improves motion binding with a relative gain of 52.46%, numeracy with a relative gain of 40.11%, and spatial relationship with a relative gain of 11.15% with CogVideoX-5B as a T2V generation backbone. We provide comprehensive quantitative and qualitative ablation studies about noise inversion ratio, different background generators, background object detection, and foreground object segmentation. We hope our method can inspire future work on effectively and efficiently integrating LLMs’ planning ability into video generation.

## 2 RELATED WORK

### 2.1 MLLM PLANNING FOR VIDEO GENERATION

There are recent research works (Wang et al., 2024c; Lin et al., 2024; Lian et al., 2024; He et al., 2023; Zhou et al., 2024b) that leverage the reasoning capabilities and world knowledge of LLMs or multimodal LLMs for the task of video generation. For example, one line of work (Lin et al., 2024; Wang et al., 2024c; Lian et al., 2024) applies GPT-4 / GPT-4o to expand a single text prompt into a ‘video plan’ in the format of bounding boxes or detailed prompt description (Yang et al., 2024a), which is then given as input to downstream video diffusion model for layout-guided video generation. The other line of work (Kondratyuk et al., 2024; Tong et al., 2024; Wang et al., 2024a; Wu et al., 2024b; Lu et al., 2023; 2024) performs token-level planning utilizing multimodal LLMs. For example, (Wang et al., 2024a; Kondratyuk et al., 2024) tokenize videos and text into the same space and generate video tokens using the same strategy as text (e.g., next-token prediction). However, both directions either rely on high-quality prompts and the bounding box planning or require extensive training and do not fully leverage the power of existing visual tools for fine-grained video generation.

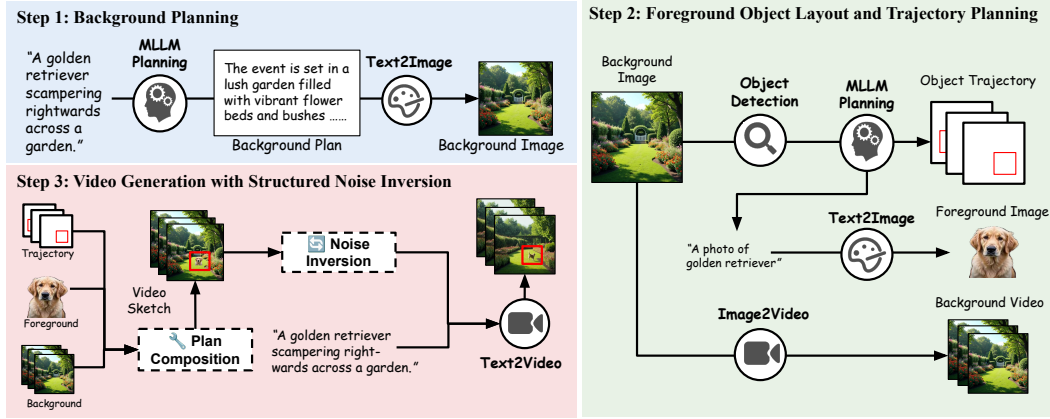


Figure 2: Three stages of VIDEO-MSG. In the first stage, the MLLM plans specific global and local contexts that fit the provided text-to-video prompt. The text-to-image (T2I) model uses the MLLM planned context to render the necessary components of the video. In the third stage, we generate video with VIDEO SKETCH via noise inversion.

In contrast, our work leverages the power of both multimodal LLMs and image/video diffusion models to generate a VIDEO SKETCH for final fine-grained motion control, and is fully training-free.

## 2.2 MOTION DIRECTION CONTROL IN VIDEO GENERATION

Controllability in video generation is gaining increasing attention in the field of generative AI, as it enables models to generate videos aligned with user intent. One line of research focuses on training models with the capability of trajectory control, camera control, or motion control by generating intermediate representations. For trajectory control, recent works such as DragNUMA (Yin et al., 2023), IVA-0 (Yu et al., 2024), DragAnything (Wu et al., 2024a), and TrackGo (Zhou et al., 2024a) encode object movement trajectories into dense features, which are then fused into the diffusion model to enable object movement control. On the other hand, CameraCtrl (He et al., 2024), MotionCtrl (Wang et al., 2024b), and Image Conductor (Li et al., 2024) encode camera extrinsics as features to control camera motion in the generated videos. A common drawback of both of these categories is their reliance on accurate object trajectory or camera movement information, which are difficult for users to manipulate directly. Additionally, video datasets with accurate trajectory annotations are limited, which constrains the performance of these models. The third category, including VideoJAM (Chefer et al., 2025) and MotionI2V (Shi et al., 2024), produces motion and video representations jointly, or sequentially by first generating intermediate representations, which then serve as guidance for generating video outputs. However, such methods require extensive training due to extra generation objectives. In contrast, our method uses an image-to-video model, allowing us to transform existing, real-world images into controllable videos under LLM planning.

## 3 METHOD

We introduce **VIDEO-MSG**, **M**ultimodal **S**ket**G**u**I**dance for video generation, a training-free guidance method for T2V generation based on multimodal planning and structured noise initialization. VIDEO-MSG consists of three stages (illustrated in fig. 2):

- **Background planning** (section 3.1), where we adopt T2I and I2V models to generate background image priors with natural animation.
- **Foreground Object Layout and Trajectory Planning** (section 3.2), where we apply MLLM and object detectors to plan and place foreground objects into the background harmoniously.
- **Video Generation with Structured Noise Initialization** (section 3.3), where the synthesized images derived from the above stages are used as VIDEO SKETCH for final video generation via inversion techniques.

### 3.1 BACKGROUND PLANNING

Given a prompt for video generation, we first ask an MLLM (GPT-4o (OpenAI et al., 2024)) to generate a detailed background description (see Stage 1 in fig. 2). Here, we explicitly instruct the MLLM to generate only the background and avoid including any moving or key objects mentioned in the original prompt, thereby enforcing proper decoupling. We find that this strategy helps address issues in conditional T2I generation based on bounding boxes, where the T2I model may fail to generate the foreground object at the specified box location in the image. In addition, we explore two approaches for background generation:

- (1) Using a T2I model to generate an initial background, followed by an I2V model to animate it. In this way, we can adopt a strong T2V model to potentially achieve improved video aesthetic quality.
- (2) Directly using a T2V model to generate the background with animation, which avoids the potential distribution gap between the two models in (1).

In both cases, we adopt a video generation model. We aim to introduce natural background animation rather than keeping it static while only animating foreground objects. This ensures that elements such as flowing water, moving clouds, or swaying trees are naturally incorporated, making the generated videos more realistic and visually coherent. Moreover, by comparing approaches (1) with (2), we notice that the advantage of adopting a strong T2I model in (1) outweighs the domain gap between the T2I and I2V models in (2) as discussed in Sec. 4.3. Therefore, we apply approach (1) as our default experiment setting.

### 3.2 FOREGROUND OBJECT LAYOUT AND TRAJECTORY PLANNING

This stage aims to place the property of the foreground object in the background in a spatially coherent manner. We first implement this stage by providing the background images generated in stage 1, along with a prompt describing movement dynamics to GPT-4o (OpenAI et al., 2024), then ask it to generate a sequence of bounding boxes to represent the foreground object’s movement. For instance, given the text prompt: “A cat sinking to the left in the living room”, GPT-4o can correctly infer the cat’s movement direction (i.e., moving left). However, when provided with a background image of a living room, GPT-4o often fails to position the cat’s bounding box appropriately on the floor (e.g., with the bounding box floating in mid-air or overlapping with unrelated objects), as illustrated in Figure 4a. This suggests that while GPT-4o demonstrates strong motion reasoning capabilities, it lacks direct grounding capability for visual elements and struggles to align foreground objects with the background scene in a spatially consistent manner.

To overcome this limitation, we first detect all objects in the background image with Recognize-Anything (RAM) (Zhang et al., 2024a) then extract their bounding boxes with Grounding-DINO (Liu et al., 2023). These bounding boxes are fed into GPT-4o to provide explicit spatial context, which helps it accurately position and animate foreground objects, enhancing spatial coherence in generated videos and reducing placement errors. Qualitative examples of the effectiveness of object detection with Grounding-DINO and RAM are presented in Figure 4a. With the above inputs (i.e., video text prompt, background image, and the bounding boxes of objects in the background), GPT-4o generates a sequence of bounding boxes for the foreground objects in the format [object name, bounding box coordinates] (see stage 2 in fig. 2). Additionally, it provides a textual description for each frame and a reasoning process explaining the planned object motions after the sequence of frames. This reasoning step enhances the coherence and accuracy of motion planning.

Once the sequence of bounding boxes is obtained, we utilize a T2I model to generate the appearance of the foreground object using the prompt: “An image of {object name}.” However, directly merging the generated object image with the background presents a challenge—the background in the generated object image can significantly affect the overall visual coherence, as illustrated in fig. 4b. To address this, we apply SAM (Kirillov et al., 2023) to extract the object from the generated image, removing any unintended background. Based on the planned bounding boxes, the extracted object is then resized and placed onto the background image at the corresponding location. This process ensures a more seamless integration of the foreground object into the background, improving the visual consistency of the generated video.

### 3.3 VIDEO GENERATION WITH STRUCTURED NOISE INITIALIZATION

In this stage, we generate a final video by guiding the T2V diffusion model with the VIDEO SKETCH created from the previous stage (section 3.2). Inversion methods (Song et al., 2020), which are often used in image and video editing tasks (Sheynin et al., 2023; Meng et al., 2022), can be effectively utilized here to create structured noise to fuse the information from VIDEO SKETCH. While the normal denoising process starts from the terminal timestep  $t = T$  (a random noise) to the initial timestep  $t = 0$  (a clean video), we create per-frame initial noises from VIDEO SKETCH via noise inversion (Meng et al., 2022) and start denoising from a timestep  $t^{\text{inv}}$ . Specifically, we first encode the sequence of VIDEO SKETCH frames into the latent space  $z$  using a 3D VAE (Yang et al., 2024b; Kingma & Welling, 2014). Next, we obtain the initial noise  $z_{t^{\text{inv}}}$  via the forward diffusion process (Ho et al., 2020):  $z_{t^{\text{inv}}} = \sqrt{\alpha_t} z_0 + \sqrt{1 - \alpha_t} \epsilon$ ,  $\epsilon \sim \mathcal{N}(0, I)$ , where  $\alpha_t = \prod_{s=1}^t (1 - \beta_s)$  is the cumulative noise schedule, and  $\epsilon$  represents Gaussian noise. We parameterize  $t^{\text{inv}} = \alpha \times T$ , where  $\alpha \in (0.0, 1.0)$ . Inspired by VideoDirectorGPT (Lin et al., 2024), which uses an LLM to estimate a confidence score along with bounding box layouts as layout guidance strength, we employ an LLM to infer an appropriate noise inversion ratio  $\alpha$  value given a text description. (see section 4.3 for detailed experiments). We explain more details about the noise inversion in Appendix.

## 4 EXPERIMENTS

### 4.1 EXPERIMENT SETUPS

**Datasets.** We evaluate VIDEO-MSG on popular text-to-video generation benchmarks, T2V-CompBench (Sun et al., 2024) and VBench (Huang et al., 2024). T2V-CompBench and VBench measure diverse aspects of text-to-video generation tasks with seven (e.g., consistent attribute binding, motion binding, spatial relationships) and sixteen categories (e.g., overall consistency, color, temporal flickering, motion smoothness), respectively. In this work, we primarily use T2V-CompBench to evaluate video diffusion models’ capability in compositional text-to-video generation, and use VBench to measure the motion smoothness of the generated video.

**Implementation details.** We implement VIDEO-MSG on two recent text-to-video generation diffusion models: VideoCrafter2 (Chen et al., 2024) and CogVideoX-5B (Yang et al., 2024b). To generate the VIDEO SKETCH, we employ FLUX.1-dev (Labs, 2024) and SDXL (Podell et al., 2024) as the background generator, and CogVideoX-5B as the image-to-video generator. We utilize Recognize-Anything (Zhang et al., 2024a) and Grounded-Segment-Anything (Kirillov et al., 2023) for foreground object segmentation. We utilize GPT-4o as the multi-modal LLM for background description generation, foreground object layout and trajectory planning, and determining the noise inversion ratio  $\alpha$  dynamically based on the prompt. For noise inversion ratio  $\alpha$  (section 3.3), we find the range [0.7, 0.9] works well for CogVideoX-5B, and the range [0.5, 0.8] works well for VideoCrafter2 (see section 4.3 for ablation study). All experiments are conducted on A100 and A6000 GPUs, with batch size 1 and an approximate memory usage of 16 GB. We provide additional details, such as prompts used for GPT-4o, in the Appendix.

### 4.2 QUANTITATIVE EVALUATION

**Improved control on spatial layout and object trajectory.** Table 1 shows that VIDEO-MSG significantly improves both T2V backbone models (VideoCrafter2 and CogVideoX-5B) in many skills, especially in motion binding (‘Motion’), with an increase of 0.1499 on VideoCrafter2 and 0.1544 on CogVideoX-5B. VIDEO-MSG also provides large improvements in spatial relationships (‘Spatial’), and numeracy (‘Numeracy’) in both backbone models. These results show that the planning and structured noise initialization of VIDEO-MSG effectively improve the control of spatial layouts and object trajectories in video generation. It is also noteworthy that VIDEO-MSG, implemented with open-source T2V backbone models, archives higher motion binding scores than closed-source models such as Gen-3 (Runway, 2024). The VIDEO-MSG did not improve the scores in dynamic attribute binding (‘Dynamic-attr’) and object action and interaction (‘Action’ and ‘Interaction’) categories. This is likely because dynamic changes in object or environment states and interactions and actions between objects are difficult to guide solely with bounding boxes.

| Model                                    | Consist-attr     | Dynamic-attr     | Spatial          | Motion           | Action           | Interaction      | Numeracy         |
|------------------------------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| <i>(Closed-source models)</i>            |                  |                  |                  |                  |                  |                  |                  |
| Pika (Team, 2024a)                       | 0.6513           | 0.1744           | 0.5043           | 0.2221           | 0.5380           | 0.6625           | 0.2613           |
| Gen-3 (Runway, 2024)                     | 0.7045           | 0.2078           | 0.5533           | 0.3111           | 0.6280           | 0.7900           | 0.2169           |
| Dreamina (Dreamina AI, 2024)             | 0.8220           | 0.2114           | 0.6083           | 0.2391           | 0.6660           | 0.8175           | 0.4006           |
| PixVerse (PixVerse, 2024)                | 0.7370           | 0.1738           | 0.5874           | 0.2178           | 0.6960           | 0.8275           | 0.3281           |
| Kling (Kling AI, 2024)                   | 0.8045           | 0.2256           | 0.6150           | 0.2448           | 0.6460           | 0.8475           | 0.3044           |
| <i>(Open-source models)</i>              |                  |                  |                  |                  |                  |                  |                  |
| ModelScope (Wang et al., 2023)           | 0.5483           | 0.1654           | 0.4220           | 0.2552           | 0.4880           | 0.7075           | 0.2066           |
| ZeroScope (Sterling, 2024)               | 0.4495           | 0.1086           | 0.4073           | 0.2319           | 0.4620           | 0.5550           | 0.2378           |
| AnimateDiff (Guo et al., 2023)           | 0.4883           | 0.1764           | 0.3883           | 0.2236           | 0.4140           | 0.6550           | 0.0884           |
| Latte (Ma et al., 2024)                  | 0.5325           | 0.1598           | 0.4476           | 0.2187           | 0.5200           | 0.6625           | 0.2187           |
| Show-1 (Zhang et al., 2023)              | 0.6388           | 0.1828           | 0.4649           | 0.2316           | 0.4940           | 0.7700           | 0.1644           |
| Open-Sora 1.2 (hpcaitech, 2024)          | 0.6600           | 0.1714           | 0.5406           | 0.2388           | 0.5717           | 0.7400           | 0.2556           |
| Open-Sora-Plan v1.1.0 (Lab & etc., 2024) | 0.7413           | 0.1770           | 0.5587           | 0.2187           | <b>0.6780</b>    | 0.7275           | 0.2928           |
| VideoTetris (Tian et al., 2024)          | 0.7125           | 0.2066           | 0.5148           | 0.2204           | 0.5280           | 0.7600           | 0.2609           |
| Vico (Yang & Wang, 2024)                 | 0.7025           | <b>0.2376</b>    | 0.4952           | 0.2225           | 0.5480           | 0.7775           | 0.2116           |
| VideoCrafter2 (Chen et al., 2024)        | 0.6750           | 0.1850           | 0.4891           | 0.2233           | 0.5800           | 0.7600           | 0.2041           |
| VideoCrafter2 + LVD (Lian et al., 2023)  | 0.6663           | 0.2308           | 0.5106           | 0.2178           | 0.5640           | 0.8125           | 0.2869           |
|                                          | <b>(-0.0087)</b> | <b>(+0.0458)</b> | <b>(+0.0215)</b> | <b>(-0.0055)</b> | <b>(-0.0160)</b> | <b>(+0.0525)</b> | <b>(+0.0828)</b> |
| VideoCrafter2 + VIDEO-MSG (Ours)         | <b>0.7536</b>    | 0.2110           | 0.5866           | 0.3732           | 0.5737           | <b>0.8220</b>    | 0.3138           |
|                                          | <b>(+0.0786)</b> | <b>(+0.0260)</b> | <b>(+0.0975)</b> | <b>(+0.1499)</b> | <b>(-0.0063)</b> | <b>(+0.0620)</b> | <b>(+0.1097)</b> |
| CogVideoX-5B (Yang et al., 2024b)        | 0.7220           | 0.2334           | 0.5461           | 0.2943           | 0.5960           | 0.7950           | 0.2603           |
| CogVideoX-5B + VIDEO-MSG (Ours)          | 0.7109           | 0.2102           | <b>0.6070</b>    | <b>0.4487</b>    | <b>0.5960</b>    | 0.7800           | <b>0.3647</b>    |
|                                          | <b>(-0.0111)</b> | <b>(-0.0232)</b> | <b>(+0.0609)</b> | <b>(+0.1544)</b> | <b>(+0.0000)</b> | <b>(-0.0150)</b> | <b>(+0.1044)</b> |

Table 1: **T2V-CompBench evaluation results.** We highlight the best/second-best scores for open-sourced models with **bold/underline**.

| No. | Noise inversion ratio $\alpha$ | T2V-CompBench  |               |               | VBench            |
|-----|--------------------------------|----------------|---------------|---------------|-------------------|
|     |                                | Motion Binding | Numeracy      | Spatial       | Motion Smoothness |
| 1.  | Direct T2V (no inversion)      | 0.2233         | 0.2041        | 0.4891        | 97.73             |
| 2.  | 0.8                            | 0.2793         | 0.2081        | 0.5502        | 98.69             |
| 3.  | 0.7                            | 0.3197         | 0.2653        | 0.5678        | 98.62             |
| 4.  | 0.6                            | 0.3352         | 0.3059        | 0.6057        | 98.63             |
| 5.  | 0.5                            | <b>0.3980</b>  | <b>0.3138</b> | <b>0.6447</b> | 98.58             |
| 6.  | LLM-controlled                 | 0.3732         | <b>0.3138</b> | 0.5866        | <b>99.01</b>      |

Table 2: Comparison of different noise inversion ratio  $\alpha$ , where we compare static values and LLM-based dynamic values. Backbone T2V: VideoCrafter2. Background generator: Flux + CogVideoX-5B.

**Comparison to planning-based baseline.** We also compare VIDEO-MSG with LVD (Lian et al., 2023), a recent T2V layout guidance method, where it adds a gradient-based energy function optimization step before each denoising step of the T2V diffusion backbone. The energy function adjusts the cross-attention map of diffusion models to concentrate within a set of object bounding boxes generated by an LLM. On the VideoCrafter2 backbone, we find that VIDEO-MSG outperforms LVD in all categories except for dynamic attribute binding, with the largest improvement observed in motion binding (‘Motion’), where VIDEO-MSG surpasses LVD by 0.1554. This demonstrates the effectiveness of our approach. Note that VIDEO-MSG is also more memory-efficient than LVD, as the layout guidance in LVD requires backpropagation through the T2V diffusion backbone, making it hard to adapt to large diffusion models; we could implement VIDEO-MSG with CogVideoX-5B backbone to run on an A6000 GPU (48GB), but we could not fit LVD even on an A100 (80GB).

### 4.3 ABLATION STUDIES

**Noise inversion ratio  $\alpha$ .** As described in section 3.3, we guide the T2V generation backbone by denoising from an intermediate timestep  $t^{\text{inv}} = \alpha \times T$  to the initial timestep  $t = 0$ . Here, we experiment with different noise inversion ratios  $\alpha$  (i.e., varying the noise injected into the VIDEO SKETCH). Table 2 shows that lower  $\alpha$  achieves better performance in motion binding (e.g., moving left/right), numeracy, and spatial relationships but hurts the smoothness of motions. This aligns with the intuition that increasing the number of refinement steps based on VIDEO SKETCH enhances the final motion quality. We observe that automatically inferring proper  $\alpha$  given text description with LLM achieves a good trade-off and use this approach by default.

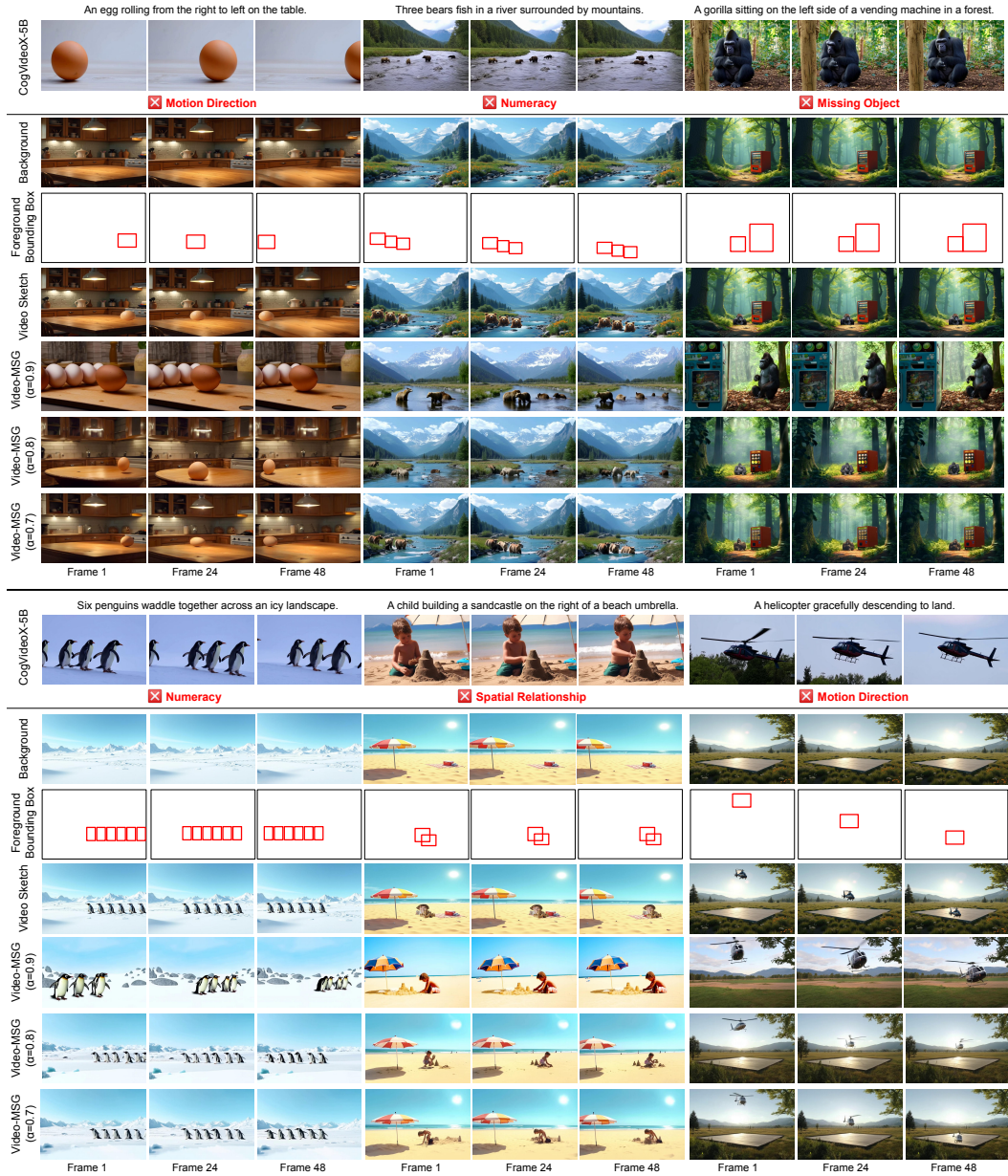


Figure 3: Videos generated with CogVideoX-5B and VIDEO-MSG with CogVideoX-5B backbone. The videos generated with VIDEO-MSG are more accurate regarding object motions, numeracy, and spatial relationships.

**Different background generator.** In Table 3, we compare different background generation methods (section 3.1): (1) generating a background with a text-to-image (T2I) model, followed by an image-to-video (I2V) model for animation, and (2) directly using a text-to-video (T2V) model to generate an animated background. While both approaches improve motion binding and numeracy compared to using a single T2V model for video generation, the T2I + I2V pipeline scores higher in numeracy, and the T2V

In Table 3, we compare different background

| No. | Background Generator            | Motion        | Numeracy      |
|-----|---------------------------------|---------------|---------------|
| 1.  | Direct T2V (no background)      | 0.2897        | 0.2750        |
| 2.  | SDXL (T2I) + CogVideoX-5B (I2V) | 0.4487        | 0.3559        |
| 3.  | FLUX (T2I) + CogVideoX-5B (I2V) | 0.4549        | <b>0.3647</b> |
| 4.  | CogVideoX-5B (T2V)              | <b>0.4565</b> | 0.3028        |

Table 3: Ablation studies on different background generators.

approach scores higher in motion binding. We attribute this to the video generation model’s ability to better refine object motion when the background follows a static camera, making foreground changes more salient for the video diffusion model. The I2V pipeline better adheres to the “Static Camera” prompt, producing natural background animations (e.g., wind, light changes). In contrast, T2V models often disregard the “Static Camera” requirement, introducing excessive camera motion and scene changes in the video. These inconsistencies make it harder for the video diffusion model to refine foreground objects, leading to performance degradation (e.g., 0.3028 with CogVideoX-5B vs. 0.3647 with FLUX on numeracy). Additionally, we find that a stronger T2I model (e.g., FLUX (Labs, 2024)) yields better results than a weaker one (e.g., SDXL (Podell et al., 2024)), highlighting the potential of leveraging high-quality T2I models for layout-controlled text-to-video generation.

**Different LLM planner.** In Table 4, we compare different LLMs for generating foreground and background plans: GPT-4o and Qwen2-VL-7B-Instruct (Bai et al., 2023). In both settings, we use VideoCrafter2 as the video generator, and 0.5 as the noise inversion ratio. Our results show that stronger LLMs produce plans that align more closely with the input prompts, and high-quality plans significantly enhance performance. Specifically, for spatial relationship, using plans generated by GPT-4o improves the baseline by 0.1556, whereas Qwen2-VL-7B-Instruct yields only a 0.0275 improvement. These results highlight the importance of plan quality and suggest that leveraging more powerful LLMs can further boost performance.

| No. | LLM Planner          | Spatial       |
|-----|----------------------|---------------|
| 1.  | Direct T2V (no LLM)  | 0.4891        |
| 2.  | Qwen2-VL-7B-Instruct | 0.5166        |
| 3.  | GPT4-o               | <b>0.6447</b> |

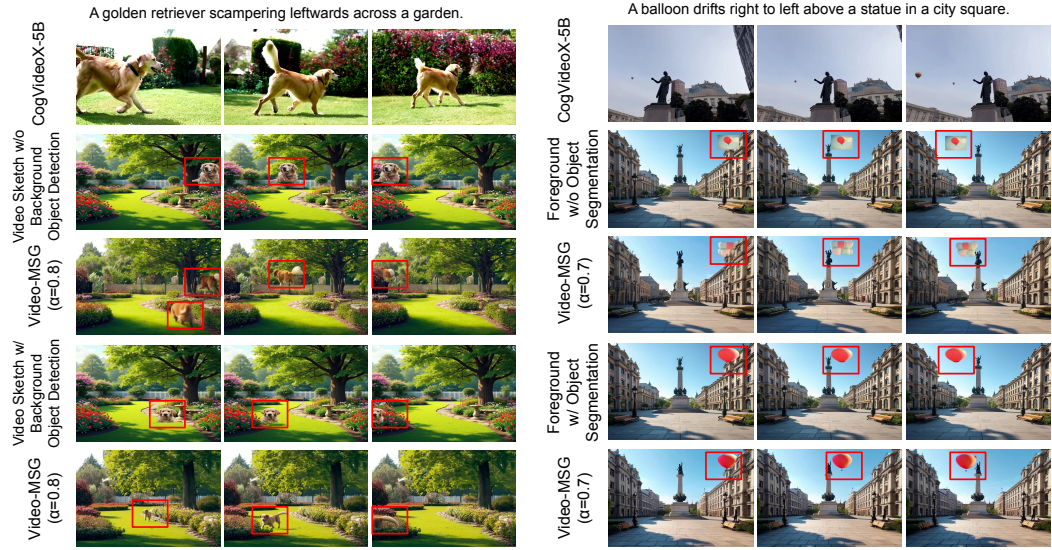
Table 4: Ablation studies on different llm planners. Backbone T2V: VideoCrafter2. Noise inversion ratio: 0.5.

#### 4.4 QUALITATIVE ANALYSIS

**VIDEO SKETCH improves control of spatial layout and object trajectory.** fig. 3 compares videos generated from CogVideoX-5B, and VIDEO-MSG (with CogVideoX-5B backbone). We observe that CogVideoX-5B struggles with motion direction (e.g., an egg moves to the right instead of to the left, a helicopter ascends instead of descending to the land), numeracy (e.g., generated four bears instead of three bears, four penguins instead of six penguins), and spatial relationships (e.g., a vending machine should be located to the right of a gorilla, but it is missing; the umbrella should be located on the left of the children). In contrast, VIDEO-MSG successfully guides the T2V backbone to generate videos with correct semantics in all cases. Note that the T2V model can understand the coarse guidance in VIDEO SKETCH and place objects that harmonize well with the background through noise inversion. For example, in the middle example (‘three bears in a river surrounded by mountains’), even when the VIDEO SKETCH includes three bears only with other heads facing forward, the T2V model could place the three bears in the river naturally.

**Effect of different noise inversion ratios  $\alpha$ .** fig. 3 shows the video generation results from VIDEO SKETCH (with CogVideoX-5B backbone), with different noise inversion ratios  $\alpha$ . Interestingly, the model can automatically refine objects to better align with the prompt and surrounding environment based on different  $\alpha$ . We find that lower  $\alpha$  (i.e., less noise) generally provides stronger layout control. For example, in the left top example, the egg in the videos with  $\alpha = 0.7$  and  $\alpha = 0.8$  closely follow the trajectory in VIDEO SKETCH, while in the video  $\alpha = 0.9$ , the egg movement is small and does not follow the trajectory. However, a lower  $\alpha$  can lead to less natural generations; e.g., in the bottom-middle example, the boy motion appears less natural at  $\alpha = 0.7$  compared to  $\alpha = 0.9$ . This highlights the importance of selecting an appropriate  $\alpha$  to balance motion smoothness with faithful adherence to VIDEO SKETCH.

**Background object detection helps foreground object placement.** We find that a deep understanding of the background images through object detection is crucial in foreground planning (section 3.2). As shown in fig. 4a, without access to background bounding box information, the MLLM fails to place the golden retriever on the grass when relying solely on the background image input. In contrast, when provided with bounding box information from the background (e.g., {"label": "path", "box": [0.44, 0.57, 0.99, 0.99]}), the MLLM successfully positions the golden re-



(a) Example video showing the importance of background object detection in foreground object placement. (b) Example video showing the importance of foreground object segmentation.

Figure 4: Qualitative examples demonstrating the importance of background object detection and foreground object segmentation.

triever at the correct location on the grass. Moreover, we find that this planning step directly impacts the final video quality. Conditioning the generation on inaccurate bounding box plans can conflict with the video diffusion model’s prior knowledge. For instance, in the first frame, the model may generate two golden retrievers—one on the ground based on its prior knowledge and another floating in the air according to the VIDEO SKETCH—resulting in unrealistic outputs, such as a golden retriever running mid-air across the garden. In contrast, our approach, which conditions planning on background bounding boxes, enables the generation of more natural and commonsense-aligned videos.

**Segmentation of foreground objects improves harmonization.** As demonstrated in fig. 4b, without object segmentation, the foreground object (a balloon) does not align well with the background, and the quantity is not well controlled (multiple balloons). This occurs because the video diffusion model does not inherently distinguish between the appearance of the background and that of the balloon, causing them to blend together. In contrast, when we first segment the balloon from the generated foreground object image and then place it onto the background to create VIDEO SKETCH, the balloon in the generated video harmonizes well with the background.

## 5 CONCLUSION

In this work, we introduce VIDEO-MSG, a training-free guidance method designed to enhance text-to-video (T2V) generation through multimodal planning and structured noise initialization. VIDEO-MSG consists of three steps, wherein the first two steps, VIDEO-MSG creates VIDEO SKETCH, a detailed spatial and temporal plan utilizing a set of multimodal models, including multimodal LLM, object detection, and instance segmentation models. In the final step, VIDEO-MSG guides a downstream T2V diffusion model with VIDEO SKETCH through noise inversion and denoising. Notably, VIDEO-MSG does not require fine-tuning or additional memory during inference, making it easier to adopt large T2V models than existing methods that rely on fine-tuning or iterative attention manipulation. VIDEO-MSG demonstrates its effectiveness in enhancing text alignment with multiple T2V backbones on popular T2V generation benchmarks. We also provide comprehensive ablation studies and qualitative examples that support the design choices of VIDEO-MSG. We hope our method can inspire future work on effectively and efficiently integrating LLMs’ planning capabilities into video generation.

## ETHICS STATEMENTS

We do not foresee any ethical implications beyond standard ethical and safety considerations that apply to AI research generally.

## REPRODUCIBILITY STATEMENT

We provide the code in the supplementary materials. All the implementation details can be found in Sec 4. All prompts used can be found in Appendix B.

## REFERENCES

- Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.
- Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, C. K. Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Shunting Zhang, Michael Suo, Phil Tillet, Xu Zhao, Eikan Wang, Keren Zhou, Richard Zou, Xiaodong Wang, Ajit Mathews, William Wen, Gregory Chanan, Peng Wu, and Soumith Chintala. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ASPLOS ’24, pp. 929–947, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703850. doi: 10.1145/3620665.3640366. URL <https://doi.org/10.1145/3620665.3640366>.
- Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.
- Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.
- Hila Chefer, Uriel Singer, Amit Zohar, Yuval Kirstain, Adam Polyak, Yaniv Taigman, Lior Wolf, and Shelly Sheynin. Videojam: Joint appearance-motion representations for enhanced motion generation in video models. *arXiv preprint arXiv:2502.02492*, 2025.
- Haixin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7310–7320, 2024.
- Dreamina AI. Dreamina. <https://dreamina.capcut.com/ai-tool/platform>, 2024.
- Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023.
- Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *CoRR*, 2024.
- Yingqing He, Menghan Xia, Haixin Chen, Xiaodong Cun, Yuan Gong, Jinbo Xing, Yong Zhang, Xintao Wang, Chao Weng, Ying Shan, et al. Animate-a-story: Storytelling with retrieval-augmented video generation. *arXiv preprint arXiv:2307.06940*, 2023.

- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *NeurIPS*, pp. 1–25, 2020. URL <http://arxiv.org/abs/2006.11239>.
- hpcaitech. Open-sora: Democratizing efficient video production for all, 2024. URL <https://github.com/hpcaitech/Open-Sora>.
- Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.
- Hedra Inc. Hedra. <https://www.hedra.com/>, 2025.
- Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. URL <https://arxiv.org/abs/1312.6114>.
- Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023.
- Kling AI. Kling. <https://klingai.com/>, 2024.
- Dan Kondratyuk, Lijun Yu, Xiuye Gu, Jose Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, Krishna Somandepalli, Hassan Akbari, Yair Alon, Yong Cheng, Joshua V. Dillon, Agrim Gupta, Meera Hahn, Anja Hauth, David Hendon, Alonso Martinez, David Minnen, Mikhail Sirotenko, Kihyuk Sohn, Xuan Yang, Hartwig Adam, Ming-Hsuan Yang, Irfan Essa, Huisheng Wang, David A Ross, Bryan Seybold, and Lu Jiang. Videopoet: A large language model for zero-shot video generation. In *Forty-first International Conference on Machine Learning*, 2024. URL <https://openreview.net/forum?id=LRkJwPIDuE>.
- Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.
- PKU-Yuan Lab and Tuzhan AI etc. Open-sora-plan, April 2024. URL <https://doi.org/10.5281/zenodo.10948109>.
- Black Forest Labs. Flux. <https://github.com/black-forest-labs/flux>, 2024.
- Yaowei Li, Xintao Wang, Zhaoyang Zhang, Zhouxia Wang, Ziyang Yuan, Liangbin Xie, Yuexian Zou, and Ying Shan. Image conductor: Precision control for interactive video synthesis. *CoRR*, 2024.
- Long Lian, Baifeng Shi, Adam Yala, Trevor Darrell, and Boyi Li. Llm-grounded video diffusion models. *arXiv preprint arXiv:2309.17444*, 2023.
- Long Lian, Baifeng Shi, Adam Yala, Trevor Darrell, and Boyi Li. Llm-grounded video diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.
- Xinyao Liao, Xianfang Zeng, Liao Wang, Gang Yu, Guosheng Lin, and Chi Zhang. Motionagent: Fine-grained controllable video generation via motion field agent. *arXiv preprint arXiv:2502.03207*, 2025.
- Han Lin, Abhay Zala, Jaemin Cho, and Mohit Bansal. Videodirectorgpt: Consistent multi-scene video generation via llm-guided planning. In *First Conference on Language Modeling*, 2024.
- Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.
- Fuchen Long, Zhaofan Qiu, Ting Yao, and Tao Mei. Videodrafter: Content-consistent multi-scene video generation with llm. *CoRR*, 2024.

- Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022.
- Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. In *The Eleventh International Conference on Learning Representations*, 2023.
- Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. Unified-io 2: Scaling autoregressive multimodal models with vision, language, audio, and action. In *CVPR*, 2024.
- Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024.
- TorchVision maintainers and contributors. Torchvision: Pytorch’s computer vision library. <https://github.com/pytorch/vision>, 2016.
- Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.
- OpenAI. Sora. <https://openai.com/sora/>, 2024.
- OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispositi, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Gierler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, Dane Sherburn, Daniel Kappler, Daniel Levin, Daniel Levy, David Carr, David Farhi, David Mely, David Robinson, David Sasaki, Denny Jin, Dev Valladares, Dimitris Tsipras, Doug Li, Duc Phong Nguyen, Duncan Findlay, Edede Oiwoh, Edmund Wong, Ehsan Asdar, Elizabeth Proehl, Elizabeth Yang, Eric Antonow, Eric Kramer, Eric Peterson, Eric Sigler, Eric Wallace, Eugene Brevdo, Evan Mays, Farzad Khorasani, Felipe Petroski Such, Filippo Raso, Francis Zhang, Fred von Lohmann, Freddie Sulit, Gabriel Goh, Gene Oden, Geoff Salmon, Giulio Starace, Greg Brockman, Hadi Salman, Haiming Bao, Haitang Hu, Hannah Wong, Haoyu Wang, Heather Schmidt, Heather Whitney, Heewoo Jun, Hendrik Kirchner, Henrique Ponde de Oliveira Pinto, Hongyu Ren, Huiwen Chang, Hyung Won Chung, Ian Kivlichan, Ian O’Connell, Ian O’Connell, Ian Osband, Ian Silber, Ian Sohl, Ibrahim Okuyucu, Ikai Lan, Ilya Kostrikov, Ilya Sutskever, Ingmar Kanitscheider, Ishaan Gulrajani, Jacob Coxon, Jacob Menick, Jakub Pachocki, James Aung, James Betker, James Crooks, James Lennon, Jamie Kiros, Jan Leike, Jane Park, Jason Kwon, Jason Phang, Jason Teplitz, Jason Wei, Jason Wolfe, Jay Chen, Jeff Harris, Jenia Varavva, Jessica Gan Lee, Jessica Shieh, Ji Lin, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joanne Jang, Joaquin Quinonero Candela, Joe Beutler, Joe Landers, Joel Parish, Johannes Heidecke, John Schulman, Jonathan Lachman, Jonathan McKay, Jonathan Uesato, Jonathan Ward, Jong Wook Kim, Joost Huizinga, Jordan Sitkin, Jos Kraaijeveld, Josh Gross, Josh Kaplan, Josh Snyder, Joshua Achiam, Joy Jiao, Joyce Lee, Juntang Zhuang, Justyn Hariman, Kai Fricke, Kai Hayashi, Karan Singhal, Katy Shi, Kavin Karthik, Kayla Wood, Kendra Rimbach, Kenny Hsu, Kenny Nguyen, Keren Gu-Lemberg, Kevin Button, Kevin Liu, Kiel Howe,

- 648 Krithika Muthukumar, Kyle Luther, Lama Ahmad, Larry Kai, Lauren Itow, Lauren Workman,  
 649 Leher Pathak, Leo Chen, Li Jing, Lia Guy, Liam Fedus, Liang Zhou, Lien Mamitsuka, Lilian Weng,  
 650 Lindsay McCallum, Lindsey Held, Long Ouyang, Louis Feuvrier, Lu Zhang, Lukas Kondraciuk,  
 651 Lukasz Kaiser, Luke Hewitt, Luke Metz, Lyric Doshi, Mada Aflak, Maddie Simens, Madelaine  
 652 Boyd, Madeleine Thompson, Marat Dukhan, Mark Chen, Mark Gray, Mark Hudnall, Marvin  
 653 Zhang, Marwan Aljubei, Mateusz Litwin, Matthew Zeng, Max Johnson, Maya Shetty, Mayank  
 654 Gupta, Meghan Shah, Mehmet Yatbaz, Meng Jia Yang, Mengchao Zhong, Mia Glaese, Mianna  
 655 Chen, Michael Janner, Michael Lampe, Michael Petrov, Michael Wu, Michele Wang, Michelle  
 656 Fradin, Michelle Pokrass, Miguel Castro, Miguel Oom Temudo de Castro, Mikhail Pavlov, Miles  
 657 Brundage, Miles Wang, Minal Khan, Mira Murati, Mo Bavarian, Molly Lin, Murat Yesildal, Nacho  
 658 Soto, Natalia Gimelshein, Natalie Cone, Natalie Staudacher, Natalie Summers, Natan LaFontaine,  
 659 Neil Chowdhury, Nick Ryder, Nick Stathas, Nick Turley, Nik Tezak, Niko Felix, Nithanth Kudige,  
 660 Nitish Keskar, Noah Deutsch, Noel Bundick, Nora Puckett, Ofir Nachum, Ola Okelola, Oleg Boiko,  
 661 Oleg Murk, Oliver Jaffe, Olivia Watkins, Olivier Godement, Owen Campbell-Moore, Patrick  
 662 Chao, Paul McMillan, Pavel Belov, Peng Su, Peter Bak, Peter Bakkum, Peter Deng, Peter Dolan,  
 663 Peter Hoeschele, Peter Welinder, Phil Tillet, Philip Pronin, Philippe Tillet, Prafulla Dhariwal,  
 664 Qiming Yuan, Rachel Dias, Rachel Lim, Rahul Arora, Rajan Troll, Randall Lin, Rapha Gontijo  
 665 Lopes, Raul Puri, Reah Miyara, Reimar Leike, Renaud Gaubert, Reza Zamani, Ricky Wang, Rob  
 666 Donnelly, Rob Honsby, Rocky Smith, Rohan Sahai, Rohit Ramchandani, Romain Huet, Rory  
 667 Carmichael, Rowan Zellers, Roy Chen, Ruby Chen, Ruslan Nigmatullin, Ryan Cheu, Saachi  
 668 Jain, Sam Altman, Sam Schoenholz, Sam Toizer, Samuel Miserendino, Sandhini Agarwal, Sara  
 669 Culver, Scott Ethersmith, Scott Gray, Sean Grove, Sean Metzger, Shamez Hermeni, Shantanu  
 670 Jain, Shengjia Zhao, Sherwin Wu, Shino Jomoto, Shirong Wu, Shuaiqi, Xia, Sonia Phene, Spencer  
 671 Papay, Srinivas Narayanan, Steve Coffey, Steve Lee, Stewart Hall, Suchir Balaji, Tal Broda, Tal  
 672 Stramer, Tao Xu, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas  
 673 Cunningham, Thomas Degry, Thomas Dimson, Thomas Raoux, Thomas Shadwell, Tianhao  
 674 Zheng, Todd Underwood, Todor Markov, Toki Sherbakov, Tom Rubin, Tom Stasi, Tomer Kaftan,  
 675 Tristan Heywood, Troy Peterson, Tyce Walters, Tyna Eloundou, Valerie Qi, Veit Moeller, Vinnie  
 676 Monaco, Vishal Kuo, Vlad Fomenko, Wayne Chang, Weiye Zheng, Wenda Zhou, Wesam Manassra,  
 677 Will Sheu, Wojciech Zaremba, Yash Patil, Yilei Qian, Yongjik Kim, Youlong Cheng, Yu Zhang,  
 678 Yuchen He, Yuchen Zhang, Yujia Jin, Yunxing Dai, and Yury Malkov. Gpt-4o system card, 2024.  
 URL <https://arxiv.org/abs/2410.21276>.
- 678 Jack Parker-Holder, Philip Ball, Jake Bruce, Vibhavari Dasagi, Kristian Holsheimer, Chris-  
 679 tos Kaplanis, Alexandre Moufarek, Guy Scully, Jeremy Shar, Jimmy Shi, Stephen Spencer,  
 680 Jessica Yung, Michael Dennis, Sultan Kenjeyev, Shangbang Long, Vlad Mnih, Harris  
 681 Chan, Maxime Gazeau, Bonnie Li, Fabio Pardo, Luyu Wang, Lei Zhang, Frederic Besse,  
 682 Tim Harley, Anna Mitenkova, Jane Wang, Jeff Clune, Demis Hassabis, Raia Hadsell,  
 683 Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 2: A large-scale founda-  
 684 tion world model, 2024. URL [https://deepmind.google/discover/blog/  
 685 genie-2-a-large-scale-foundation-world-model/](https://deepmind.google/discover/blog/genie-2-a-large-scale-foundation-world-model/).
- 686 PixVerse. Pixverse. <https://app.pixverse.ai>, 2024.
- 687
- 688 Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe  
 689 Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image  
 690 synthesis. In *The Twelfth International Conference on Learning Representations*, 2024.
- 691
- 692 Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv  
 693 Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, David Yan, Dhruv Choudhary, DingKang  
 694 Wang, Geet Sethi, Guan Pang, Haoyu Ma, Ishan Misra, Ji Hou, Jialiang Wang, Kiran Jagadeesh,  
 695 Kunpeng Li, Luxin Zhang, Mannat Singh, Mary Williamson, Matt Le, Matthew Yu, Mitesh Kumar  
 696 Singh, Peizhao Zhang, Peter Vajda, Quentin Duval, Rohit Girdhar, Roshan Sumbaly, Sai Saketh  
 697 Rambhatla, Sam Tsai, Samaneh Azadi, Samyak Datta, Sanyuan Chen, Sean Bell, Sharadh  
 698 Ramaswamy, Shelly Sheynin, Siddharth Bhattacharya, Simran Motwani, Tao Xu, Tianhe Li,  
 699 Tingbo Hou, Wei-Ning Hsu, Xi Yin, Xiaoliang Dai, Yaniv Taigman, Yaqiao Luo, Yen-Cheng  
 700 Liu, Yi-Chiao Wu, Yue Zhao, Yuval Kirstain, Zecheng He, Zijian He, Albert Pumarola, Ali  
 701 Thabet, Arsiom Sanakoyeu, Arun Mallya, Baishan Guo, Boris Araya, Breana Kerr, Carleigh  
 Wood, Ce Liu, Cen Peng, Dimitry Vengertsev, Edgar Schonfeld, Elliot Blanchard, Felix Juefei-  
 Xu, Fraylie Nord, Jeff Liang, John Hoffman, Jonas Kohler, Kaolin Fire, Karthik Sivakumar,

- Lawrence Chen, Licheng Yu, Luya Gao, Markos Georgopoulos, Rashel Moritz, Sara K. Sampson, Shikai Li, Simone Parmeggiani, Steve Fine, Tara Fowler, Vladan Petrovic, and Yuming Du. Movie gen: A cast of media foundation models. <https://ai.meta.com/blog/movie-gen-media-foundation-models-generative-ai-video/>, 2025. URL <https://arxiv.org/abs/2410.13720>.
- Runway. Gen-3. <https://runwayml.com/blog/introducing-gen-3-alpha/>, 2024.
- Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. In *CVPR*, 2023. URL <https://api.semanticscholar.org/CorpusID:265221391>.
- Xiaoyu Shi, Zhaoyang Huang, Fu-Yun Wang, Weikang Bian, Dasong Li, Yi Zhang, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, et al. Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling. In *SIGGRAPH (Conference Paper Track)*, 2024.
- Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- Spencer Sterling. Zeroscope v2 576w. [https://huggingface.co/cerspense/zeroscope\\_v2\\_576w](https://huggingface.co/cerspense/zeroscope_v2_576w), 2024.
- Kaiyue Sun, Kaiyi Huang, Xian Liu, Yue Wu, Zihan Xu, Zhenguo Li, and Xihui Liu. T2v-compbench: A comprehensive benchmark for compositional text-to-video generation. *CoRR*, abs/2407.14505, 2024. URL <https://doi.org/10.48550/arXiv.2407.14505>.
- Pika Team. Pika art. <https://pika.art/>, 2024a.
- Wan Team. Wan: Open and advanced large-scale video generative models, 2025.
- WorldLab Team. Worldlab. <https://www.worldlabs.ai/blog>, 2024b.
- Ye Tian, Ling Yang, Haotian Yang, Yuan Gao, Yufan Deng, Jingmin Chen, Xintao Wang, Zhaochen Yu, Xin Tao, Pengfei Wan, Di Zhang, and Bin Cui. Videotetris: Towards compositional text-to-video generation, 2024. URL <https://arxiv.org/abs/2406.04277>.
- Shengbang Tong, David Fan, Jiachen Zhu, Yunyang Xiong, Xinlei Chen, Koustuv Sinha, Michael Rabbat, Yann LeCun, Saining Xie, and Zhuang Liu. Metamorph: Multimodal understanding and generation via instruction tuning. *arXiv preprint arXiv:2412.14164*, 2024.
- Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. <https://github.com/huggingface/diffusers>, 2022.
- Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023.
- Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *CoRR*, 2024a.
- Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *SIGGRAPH (Conference Paper Track)*, 2024b.
- Zun Wang, Jialu Li, Han Lin, Jaehong Yoon, and Mohit Bansal. Dreamrunner: Fine-grained storytelling video generation with retrieval-augmented motion adaptation. *arXiv preprint arXiv:2411.16657*, 2024c.

- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/2020.emnlp-demos.6>.
- Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting Gao, and Di Zhang. Draganything: Motion control for anything using entity representation. In *European Conference on Computer Vision*, pp. 331–348. Springer, 2024a.
- Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, et al. Vila-u: a unified foundation model integrating visual understanding and generation. *arXiv preprint arXiv:2409.04429*, 2024b.
- Ling Yang, Zhaochen Yu, Chenlin Meng, Minkai Xu, Stefano Ermon, and CUI Bin. Mastering text-to-image diffusion: Recaptioning, planning, and generating with multimodal llms. In *Forty-first International Conference on Machine Learning*, 2024a.
- Xingyi Yang and Xinchao Wang. Compositional video generation as flow equalization, 2024.
- Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024b.
- Shengming Yin, Chenfei Wu, Jian Liang, Jie Shi, Houqiang Li, Gong Ming, and Nan Duan. Dragnuwa: Fine-grained control in video generation by integrating text, image, and trajectory. *arXiv preprint arXiv:2308.08089*, 2023.
- Shoubin Yu, Jacob Zhiyuan Fang, Jian Zheng, Gunnar Sigurdsson, Vicente Ordonez, Robinson Piramuthu, and Mohit Bansal. Zero-shot controllable image-to-video animation via motion decomposition. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 3332–3341, 2024.
- David Junhao Zhang, Jay Zhangjie Wu, Jia-Wei Liu, Rui Zhao, Lingmin Ran, Yuchao Gu, Difei Gao, and Mike Zheng Shou. Show-1: Marrying pixel and latent diffusion models for text-to-video generation. *arXiv preprint arXiv:2309.15818*, 2023.
- Youcai Zhang, Xinyu Huang, Jinyu Ma, Zhaoyang Li, Zhaochuan Luo, Yanchun Xie, Yuzhuo Qin, Tong Luo, Yaqian Li, Shilong Liu, et al. Recognize anything: A strong image tagging model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1724–1732, 2024a.
- Zhenghao Zhang, Junchao Liao, Menghao Li, Zuozhuo Dai, Bingxue Qiu, Siyu Zhu, Long Qin, and Weizhi Wang. Tora: Trajectory-oriented diffusion transformer for video generation. *arXiv preprint arXiv:2407.21705*, 2024b.
- Haitao Zhou, Chuang Wang, Rui Nie, Jinxiao Lin, Dongdong Yu, Qian Yu, and Changhu Wang. Trackgo: A flexible and efficient method for controllable video generation. *CoRR*, 2024a.
- Yupeng Zhou, Daquan Zhou, Ming-Ming Cheng, Jiashi Feng, and Qibin Hou. Storydiffusion: Consistent self-attention for long-range image and video generation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b.

## APPENDIX

In this appendix, we present the following:

- Noise inversion details using DPM-Solver++ (Lu et al., 2022) in Sec. A.
- MLLM prompts we use to collect background description, foreground object layout and trajectory, and  $\alpha$  used to determine how much noise to inject during inversion in Sec. B.
- Limitations, broader impact and licenses are discussed in Sec. C and Sec. D.

You need to provide a detailed description of the background for a given event. The description should be about the environment, lighting, and setting, without including any moving objects or entities. The background description should not include any key objects that are mentioned in the prompt (e.g., don't include objA/objB if the prompt is like objA is left to objB, a white objA beside a green objB). The background description should not exceed 50 words.

Example: {In-Context Learning Example}

User: Provide a background description for the event: {Prompt}

Figure 5: Prompt template used to query background description.

## A NOISE INVERSION DETAILS

Here, we describe in detail how we adopt the inversion technique for final video generation with the motion priors prepared in Stage 3. Specifically, we first encode the sequence of images with the planned layout, collected in the previous stage, into the latent space  $z$  using a 3D Variational Autoencoder (3D VAE). Then, we perform the forward diffusion process where Gaussian noise is gradually added to the latent. Following the DPM-Solver++ (Lu et al., 2022) scheduler in CogVideoX, the noised latent at diffusion step  $t$  is:

$$z_t = \sqrt{\alpha_t} z_0 + \sqrt{1 - \alpha_t} \epsilon, \quad \epsilon \sim \mathcal{N}(0, I). \quad (1)$$

Here,  $\alpha_t = \prod_{s=1}^t (1 - \beta_s)$  is the cumulative noise schedule, and  $\epsilon$  represents Gaussian noise.

Then, given a noisy latent  $z_t$ , we attempt to recover the clean latent as a general reverse denoising starting from step  $t$ . The model then denoises to  $z_{t-1}$  using the DPM-Solver++ method, which provides a high-order approximation of the reverse diffusion process. Specifically, the update equation for  $z_{t-1}$  in a second-order solver is:

$$z_{t-1} = z_t + \lambda_1 \hat{F}(z_t, t) + \lambda_2 \hat{F}(z_t + \lambda_3 \hat{F}(z_t, t), t_m), \quad (2)$$

where  $\hat{F}(z, t) = -\frac{1}{2} \beta_t z - g^2(t) \epsilon_\theta(z, t)$  is the estimated drift term,  $\lambda_1, \lambda_2, \lambda_3$  are step-size coefficients computed adaptively, and  $t_m$  is an intermediate timestep between  $t$  and  $t - 1$ .

We observe that selecting  $t$  within a specific range enables video diffusion models to inject smooth object motions naturally. This process effectively transforms a sequence of static images into a coherent video with realistic motion dynamics.

## B PROMPT FOR MLLM PLANNING

In this section, we present the prompt templates used to collect background descriptions, foreground object layouts and trajectories, and the parameter  $\alpha$ , which determines the amount of noise injected during inversion. As shown in fig. 5, we explicitly instruct the multi-modal LLM to separate the generation of foreground and background, ensuring that key foreground objects are not mistakenly included in the background. In fig. 6, we first prompt the MLLM to generate bounding boxes for foreground objects, followed by reasoning for their placement. Additionally, we emphasize that the placement of foreground objects should be informed by background bounding box annotations to improve spatial coherence. Finally, fig. 7 illustrates the prompt template used to determine the appropriate level of noise injection during inversion. We explicitly incorporate prior knowledge into the template, instructing the MLLM to apply less noise for tasks requiring precise trajectory or layout control and more noise for tasks involving dynamic changes or object actions that cannot be effectively modeled with bounding box plans.

## C LIMITATIONS AND BROADER IMPACT

Text-to-video generation models have broad real-world applications, such as filmmaking or enhancing existing videos through recapturing. Our proposed method, VIDEO-MSG, enables improved control

Assuming the frame size is normalized to the range 0-1, you need to give a possible 25-frame layout with bounding boxes of the entities of a given event. The background image is provided. Besides, the objects in the background image are also annotated with bounding box, and these objects are static objects across the 25 frames. You need to generate the moving object at the correct location based on the input background image and bounding box annotation. You don't need to include the box plan of the objects that already in the background.

You need to generate plan for all the key objects in the prompt (e.g., Generate both objA and objB for prompt objA is left to objB). There should be enough distance between the two objects so that each object can be generated clearly.

The object name should contain its attributes (e.g., color, shape).

Number the objects if there the multiple same objects in the prompt (e.g., 2 birds).

Each object in the image is one rectangle or square box in the layout and size of boxes should be as large as possible. You need to generate layouts from the close up camera view of the event.

The layout difference between two adjacent frames must be small, considering the small interval.

You need to generate a caption that best describes the image for each frame. After generating all frames, add reasoning to your design.

Please strictly follow the user format here to generate the plan. Do not include any code or calculation before generating the plan. Add all the reasoning process after generating the plan.

Use format:

Frame\_1: [[object1, [left, top, right, bottom]], [object2, [left, top, right, bottom]], ..., [object\_n, [left, top, right, bottom]]], caption:...

Frame\_2: [[object1, [left, top, right, bottom]], [object2, [left, top, right, bottom]], ..., [object\_n, [left, top, right, bottom]]], caption:...

...

Frame\_25: [[object1, [left, top, right, bottom]], [object2, [left, top, right, bottom]], ..., [object\_n, [left, top, right, bottom]]], caption:...

Reasoning:...

Here's one example:

{In-Context Learning Example}

User Input:

User: Provide bounding box coordinates for the prompt: {Prompt}

Background box annotation: {Background Bounding Box Annotation}

Background image: {Image}

Figure 6: Prompt template used to query foreground object layout and trajectory plan.

You are an assistant tasked with guiding video generation. The video is created based on conditions set by a pre-planned sequence of images. Your role is to determine how much influence these pre-planned images should have on the final video, balanced against the video generation system's own capabilities.

Decision Rules:

1. High control by pre-planned images:
  - (1) When the prompt including the description of the movement direction of the objects.
  - (2) When the prompt emphasizes generating multiple objects in the video.
2. Low control by pre-planned images:
  - (1) When the prompt describes specific actions of objects (e.g., singing), or interaction between the objects, or focuses on the state changes of the background.
  - (2) When the prompt specifies attributes of objects (e.g., color, shape).

Control Levels: You will select one of the following control levels for the pre-planned images:

699: Maximum control by pre-planned images.

799: High control by pre-planned images.

899: Moderate control by pre-planned images.

999: Minimal control by pre-planned images.

Here're some examples:

{In-Context Learning Example}

User Input: {Prompt}

Figure 7: Prompt template used to determine how much noise to inject during inversion.

over text-to-video generation in a zero-shot setting, allowing users to more precisely direct the motion of objects within the video.

However, VIDEO-MSG has several limitations. It relies on a strong instruction-following LLM to generate planning sequences from user input. While it can be implemented using publicly available models such as Qwen2-VL, stronger models like GPT-4o yield better performance. Besides, although VIDEO-MSG improves text-to-video generation in terms of object motion, spatial relationships, and numeracy, it does not guarantee strict adherence to every detail in the text prompt. Therefore, careful evaluation is necessary before applying VIDEO-MSG in domain-specific scenarios.

## D LICENSES

We provide the licenses of the existing assets we use in this paper in table 5.

Table 5: A list of the licenses of the existing assets used in this paper.

| Asset                                          | License                                 |
|------------------------------------------------|-----------------------------------------|
| PyTorch (Ansel et al., 2024)                   | BSD-style                               |
| Huggingface Transformers (Wolf et al., 2020)   | Apache License 2.0                      |
| Torchvision (maintainers & contributors, 2016) | BSD 3-Clause “New” or “Revised” License |
| Diffusers (von Platen et al., 2022)            | Apache License 2.0                      |
| VideoCrafter2 (Chen et al., 2024)              | Apache License 2.0                      |
| CogVideoX (Yang et al., 2024b)                 | Apache License 2.0                      |
| GPT4-o (OpenAI et al., 2024)                   | OpenAI Terms of Use                     |
| Qwen2-VL (Bai et al., 2023)                    | Tongyi Qianwen LICENSE AGREEMENT        |