AdDirector: Anchored Guidance for Generating Camera-Controllable Advertisement Videos

Shiyue Zhang, Zheng Chong, Xiangkun Shi, Xiaojian Lin, Junwen Pan, Nan Chen, Cheng Chen, Chang Liu, Hanhui Li, Xiaodan Liang*

Shenzhen Campus of Sun Yat-sen University

Abstract

Reference-to-video (R2V) generation has advanced significantly and is now a promising paradigm for controllable video synthesis. Nevertheless, in advertisement video generation, where accurate camera control and precise product presentation are essential, existing approaches still face two fundamental challenges. First, current models adhere weakly to high-level semantic camera instructions, yielding stochastic camera motion. Second, they struggle to preserve strict cross-frame appearance consistency, often leading to identity-ambiguous artifacts. Although certain image animation methods incorporate explicit motion conditions (e.g., trajectories or motion fields) to improve controllability, these conditions are costly to annotate, thereby constraining their practical applicability. To address these limitations, we propose a camera-aware R2V framework named AdDirector, which adaptively exploits anchored guidance from the reference image and propagates it across spatial, temporal, and scale dimensions. Specifically, AdDirector introduces **Spatio-Temporal-Scale (STS)** modulation, a camera-conditioned modulation mechanism that first selects multi-scale reference features, and then regulates feature injection throughout the R2V generative process spatio-temporally conditioned on camera types. Furthermore, AdDirector proposes **Temporal Anchoring Regularization (TAR)**, a boundary-conditioned feature-level self-distillation scheme. TAR leverages latent trajectories extracted from a homologous model as temporal anchors, thereby reducing feature drift and improving cross-frame appearance consistency. These two modules together transform static reference conditioning into structured anchored guidance, and experiments show that our method substantially improves camera controllability, appearance consistency, and spatio-temporal coherence for advertisement video generation.

1 Introduction

The rapid advancement of generative artificial intelligence has fundamentally transformed the landscape of video content creation, particularly in the domain of reference-to-video (R2V) generation [14, 16, 23, 24, 27, 29, 33, 37, 41, 63]. Recent breakthroughs in diffusion-based video synthesis [14, 16, 23, 24, 27, 33, 37, 63] have demonstrated remarkable capabilities in generating high-quality videos from textual prompts and reference images.

However, unlike general-purpose video generation, advertisement video synthesis must simultaneously follow high-level camera instructions and faithfully preserve product appearance throughout the shot. These two requirements are tightly coupled, yet current R2V methods address neither adequately, exposing two key challenges (Fig. 1). First, **current models adhere weakly to high-level semantic camera instructions, yielding stochastic camera motion.** Although recent R2V models [23, 33, 37]

*Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

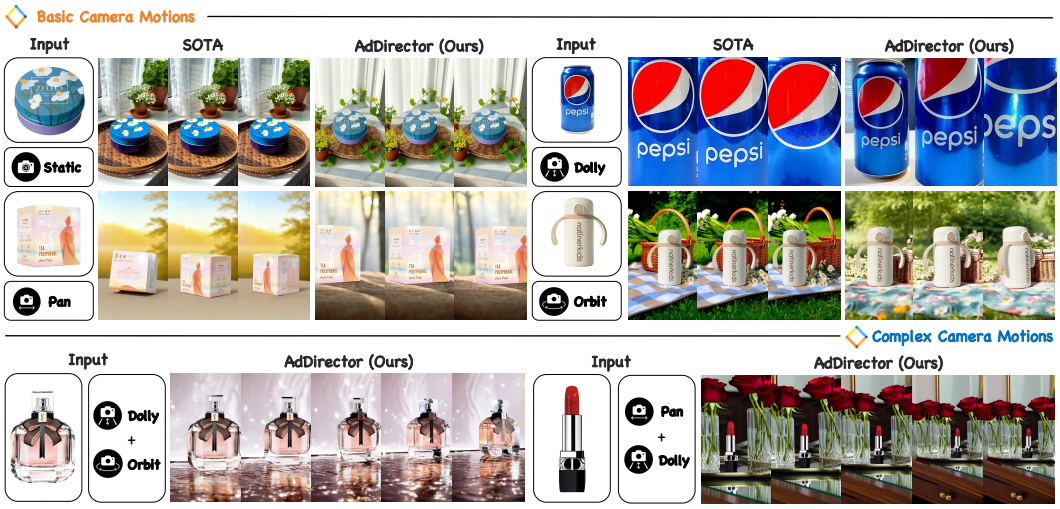

Figure 1: AdDirector demonstrates superior performance in generating camera-controllable videos while preserving high product fidelity. In contrast, existing methods struggle to maintain strict cross-frame appearance consistency (e.g., the first row) or achieve precise camera control purely from semantic camera instructions (e.g., the second row). By composing these basic camera motions, our method further enables complex camera control (e.g., the third row).

leverage large language models (LLMs) to encode detailed textual prompts, they still struggle to follow high-level camera instructions (e.g., static, dolly, orbit, and pan), yielding highly stochastic and poorly-controllable motion. This limitation stems from the weak alignment between semantic language instructions and the inherently geometric nature of camera motion: unlike scene-content generation, translating abstract camera intent into coherent spatio-temporal transformations remains difficult for current diffusion-based models. Second, **they struggle to preserve strict cross-frame appearance consistency, often leading to identity-ambiguous artifacts.** Owing to the lack of explicit supervision over latent temporal evolution, current R2V models have limited ability to preserve the geometric structure, fine-grained texture, and cross-frame consistency of reference products, producing identity-ambiguous artifacts such as logo drifting, texture re-synthesis, and geometric deformation. A further consequence is that camera motion continuously changes the product's visible region, relative scale, and demand for local detail, so a single, fixed-scale reference cannot adequately serve different camera-motion stages. Certain animation methods [10, 17, 19, 20, 45, 51, 53, 54] introduce explicit trajectory modeling, but they typically require source videos or first-frame pose annotations, together with complex geometric reasoning or manual labeling, which limits their practicality for advertisement video production.

To address this gap, we propose a camera-aware end-to-end R2V framework that synthesizes advertisement videos from a single reference image and a text prompt with camera instructions, without requiring additional control signals. Our method consists of two tightly coupled core components: **Spatio-Temporal-Scale (STS)** Modulation and **Temporal Anchoring Regularization (TAR)**, which jointly establish a generation–regularization complementary framework. **STS** is designed to handle diverse camera patterns while improving reference utilization. Specifically, we first construct a multi-scale candidate set from the reference image via image augmentation, including cropping and downsampling. We then introduce learnable 3D tokens through a reference scheduler that jointly conditions on textual camera instructions and multi-scale candidates. These 3D tokens serve as compact spatiotemporal carriers that map high-level camera semantics to stage-wise feature modulation, enabling effective reference-feature utilization with affordable computational overhead. Under a joint scheduling strategy implemented via segmented self-attention across the spatial, temporal, and scale dimensions, our framework dynamically adjusts both scale selection and feature-injection strength across generation stages. In addition, inspired by the recent progress in self-distilled LLM reasoning [1, 62], **TAR** is introduced to facilitate stable feature learning under varying camera motions. It first builds a structurally aligned anchor branch conditioned on the first and last frames with lightweight LoRA adaptation. During optimization of the main generation branch, the anchor branch remains

frozen and supervises intermediate DiT features, providing stable temporal anchors for intermediate frames and reducing instability caused by stochastic camera motion and feature drift. Overall, STS enlarges the model's representational capacity through adaptive multi-scale conditioning, while TAR regularizes that expressive space with boundary-conditioned temporal anchoring, so that generation better follows the intended camera motion while remaining temporally stable. As there is no public benchmark tailored to camera-controllable advertisement video generation, we construct a dedicated dataset for this task. On this dataset, our framework consistently outperforms strong R2V baselines across quantitative metrics, qualitative comparisons, and comprehensive user studies.

Our contributions are summarized as follows:

- We present a camera-aware end-to-end R2V framework that delivers accurate camera motion and strong product fidelity without auxiliary control signals. Extensive experiments show that our framework significantly improves camera accuracy and visual consistency over prior R2V methods.
- We introduce Spatio-Temporal-Scale (STS) Modulation, which injects multi-scale reference features under a learnable adaptive scheduling strategy, achieving a better balance between camera-motion controllability and fine-grained detail preservation.
- We propose Temporal Anchoring Regularization (TAR), which uses a frozen self-distilled branch to provide stable intermediate supervision, improving temporal appearance consistency and constraining cross-frame motion transitions.

2 Related Work

2.1 Reference-to-Video Generation

Reference-to-video (R2V) generation aims to generate videos that follow prompts while preserving the identity of reference images. Benefiting from advances in video generation models [5, 8, 18, 21, 30, 49, 56], recent R2V models [6, 9, 14, 16, 23, 24, 27, 33, 37, 63] have substantially improved identity fidelity and controllability. Existing methods include VACE [27], which structures inputs via a Video Condition Unit and encodes spatiotemporal information with a Context Adapter; Phantom [37] and HunyuanCustom [23], which enhance identity alignment via text-image fusion; MAGREF [14], which uses masked guidance and subject disentanglement; BindWeave [33], which reasons over multimodal subject interactions with MLLM; SkyReels-A2 [16], which injects multi-element features through joint embeddings. Despite this progress, existing methods still struggle with cross-frame appearance consistency and fail to faithfully follow high-level camera instructions.

2.2 Camera Control in Video Generation

Recent camera-control methods [3, 10, 17, 19, 20, 22, 43, 45, 51, 53, 54, 60] explicitly model camera trajectories to enable controllable video synthesis. Motion Prompting [17] conditions generation on 2D motion trajectories, while CameraCtrl [19] encodes per-frame camera parameters using Plücker embeddings. CAMVERSE [51] aligns trajectories with 3D geometry rewards, and Perception-as-Control [10] unifies camera and motion through 3D representations. Although these methods achieve complex camera movements, they often rely on additional inputs such as first frames, pose annotations, or 3D reconstruction, limiting their applicability to advertisement video generation.

2.3 Self-Distillation

Recent self-distillation methods [62] often construct an asymmetric teacher-student pair from the same model, where the teacher is conditioned on richer contexts, while the student preserves the inference-time interface and receives fine-grained supervision. This asymmetric paradigm has been widely explored in LLMs through on-policy self-distillation, where privileged teacher context guides student reasoning trajectories [62]. In visual diffusion, bidirectional or multi-step teachers supervise causal or few-step students [36, 55, 59]; self-distillation transfers implicit video priors into explicit 3D representations [2]. Representation-level alignment improves generative training and temporal coherence [15, 31]. Keyframe-conditioned generation further shows that first and last frames provide strong boundary context for coherent video synthesis [50, 57], but using them at inference requires extra inputs. Motivated by this self-distillation principle, TAR uses a frozen boundary-conditioned branch as the teacher to provide block-level temporal features for DiTs during training.

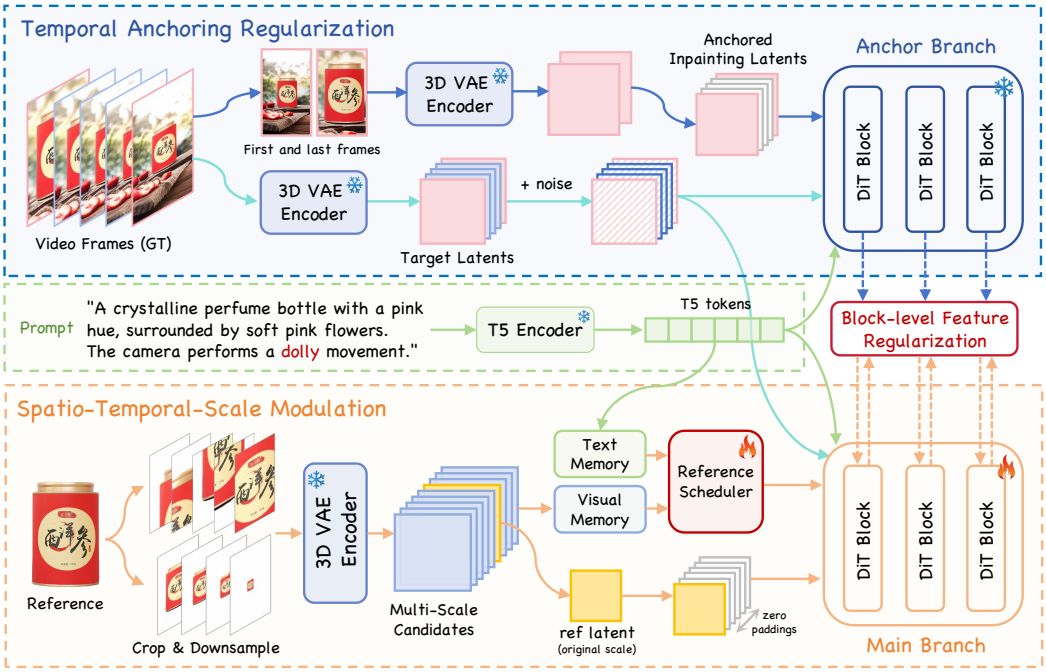

Figure 2: Overview of AdDirector. (a) It has a trainable main branch that incorporates STS modulation to construct multi-scale reference candidates and dynamically schedules features by jointly considering textual semantics and camera instructions. (b) During the training stage, block-level hidden features from a frozen anchor branch (with the same architecture as the main branch) are used as temporal targets to regularize the corresponding DiT blocks in the main branch.

3 Method

3.1 Overall Framework

The framework of AdDirector is shown in Fig. 2, which improves a base model (e.g., Wan [11]) in camera control and identity preservation via spatio-temporal-scale (STS) modulation and temporal anchoring regularization (TAR). Specifically, given a product image I as the reference and a text prompt P describing the target camera motion, our goal is to generate a video $V \in \mathbb{R}^{H\times W\times T}$, where H, W, and T denote the frame height, width, and number, respectively. To this end, the base model first employs two encoders to convert inputs into latent representations, namely, the image latent $\mathbf{y}^{\text{ref}} \in \mathbb{R}^{C_l\times 1\times H_l\times W_l}$ via the VAE encoder and the text tokens $\mathbf{p} \in \mathbb{R}^{N_t\times C_t}$ via the T5 encoder [13]. Here, C_l and C_t denote channel dimensions, while H_l and W_l denote latent spatial dimensions, and N_t denotes the number of text tokens. Following the common practice of formulating R2V as video inpainting [11], an inpainting latent $\mathbf{z} \in \mathbb{R}^{(C_l+4)\times(T_l+1)\times H_l\times W_l}$ is constructed by concatenating $\mathbf{y}^{\text{ref}}$ with an all-zero sequence and extending the channel dimension with four additional channels to encode mask information. The inpainting latent $\mathbf{z}$ is then concatenated with noise, yielding a noise latent $\mathbf{x} \in \mathbb{R}^{(2\times C_l+4)\times(T_l+1)\times H_l\times W_l}$ for subsequent denoising by several DiT blocks.

Nevertheless, the base generation process conditions on the reference image uniformly, ignoring that different camera motions emphasize different regions and detail scales at different timesteps. This mismatch makes it difficult to balance camera controllability and identity preservation. To address this issue, AdDirector introduces STS modulation to construct and fuse multi-scale reference images, producing reference latents that retain sufficient detail with moderate computational overhead. STS then injects these latents adaptively via segmented self-attention. To explicitly supervise temporal dynamics in noisy latents, TAR uses frozen DiT blocks from the base model as an anchor branch and encodes the first and last frames of each training video as temporal anchors. These anchors provide reliable latent boundary constraints that guide optimization. Details of STS modulation and TAR are presented in the following sections.

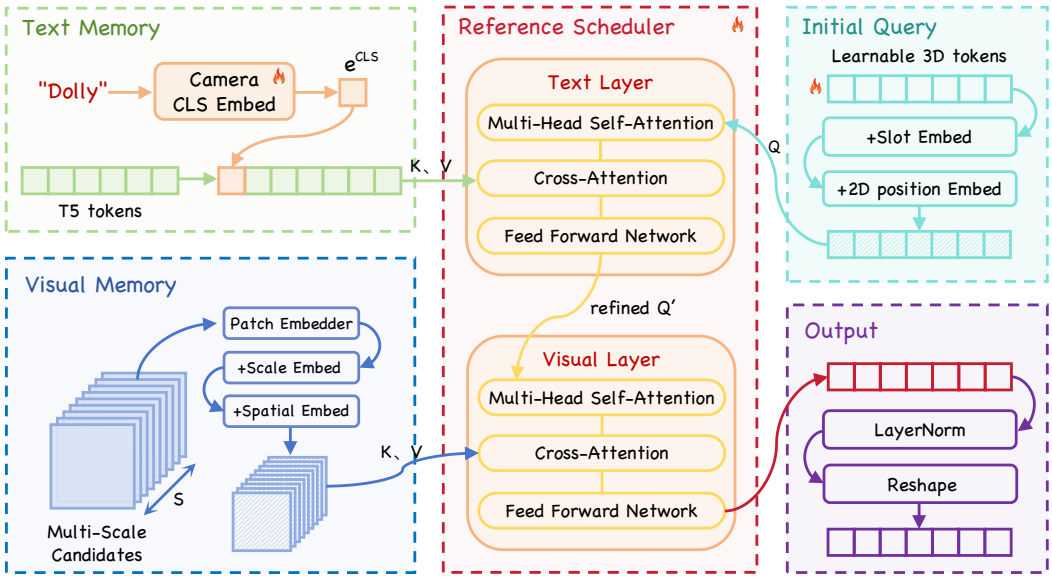

Figure 3: Illustration of the Spatio-Temporal-Scale (STS) modulation. The reference scheduler adopts a Q-Former architecture comprising two cross-attention blocks. The first block derives camera-aware semantic queries from text tokens augmented with a camera class embedding, while the second further refines these queries by attending to multi-scale reference latents. The resulting 3D tokens serve as compact spatio-temporal carriers, facilitating adaptive feature modulation within the main generation branch.

3.2 Spatio-Temporal-Scale Modulation

In advertisement video generation, varying camera motions dictate distinct requirements for spatial regions and detail scales. Conventional reference injection methods treat the reference as a static, fixed-resolution conditioning signal, neglecting two critical factors: (i) Different camera types impose unique demands on structural fidelity; for instance, zoom-in shots prioritize sharp local details, whereas orbit shots demand globally consistent textures. (ii) Injecting single-scale reference features uniformly across all temporal positions inevitably conflates global structure with local details. Therefore, our STS modulation integrates multi-scale candidate construction, a reference scheduler, and segmented self-attention to adaptively regulate guidance injection, as shown in Fig. 3.

Multi-Scale Candidate Construction. As the reference image is usually high-resolution, we construct a set of multi-scale candidates by downsampling and cropping it at different scales. Each downsampled or cropped image is then padded to the original size for unified processing. Let S denote the number of generated candidates. We encode all candidates using the VAE encoder of the base model, yielding a multi-scale reference latent set $\mathcal{Y} = \{\mathbf{y}_s\}_{s=1}^{S}$. This reference set preserves the product information across varying scales and regions.

Reference Scheduler. A naive strategy using all multi-scale candidates in $\mathcal{Y}$ will incur prohibitive computational overhead, e.g., the computational complexity of a single self-attention block will increase from $O((H_l W_l)^2)$ to $O(S^2(H_l W_l)^2)$. Therefore, we construct a learnable reference scheduler to select and summarize informative reference features. The reference scheduler adopts a lightweight Q-Former architecture [32] that mainly consists of two cross-attention blocks to generate adaptive queries from both the text tokens $\mathbf{p}$ and the multi-scale reference latent set $\mathcal{Y}$, as shown in Fig. 3. Specifically, the role of the first cross-attention block is to model camera semantics from the given prompt, which is achieved by treating $\mathbf{p}$ as the keys and values, and a sequence of learnable tokens as queries to perform cross-attention. Let $\mathbf{q} \in \mathbb{R}^{N \cdot H_h \cdot W_h \times C_h}$ denote the learnable queries, where H_h, W_h, and C_h denote the feature dimensions of the reference image in the hidden space of the DiTs. $N \ll S$ denotes the number of slots of a video, which we introduce to achieve finer camera control. Before being fed into the reference scheduler, $\mathbf{q}$ is added with a slot embedding $\mathbf{e}^{\text{slot}} \in \mathbb{R}^{C_h}$ and a positional embedding $\mathbf{e}^{\text{q-pos}} \in \mathbb{R}^{N \cdot H_h \cdot W_h \times C_h}$, to encode spatial and slot information. As for $\mathbf{p}$, we

also project it into the hidden space of the DiTs and concatenate it with a camera class embedding $\mathbf{e}^{\mathrm{CLS}} \in \mathbb{R}^{C_h}$, yielding an augmented latent $\mathbf{p}' \in \mathbb{R}^{(N_t+1)\times C_h}$. The first cross-attention block of the reference scheduler can be formulated as:

$$\mathbf{q}' = \phi(\mathbf{q}, \mathbf{p}', \mathbf{p}'), \tag{1}$$

where $\phi(Q, K, V)$ represents the standard cross-attention operation with Q, K, V as queries, keys, and values, respectively.

The second cross-attention block further refines $\mathbf{q}'$ by attending to multi-scale candidate latents. As in [49], we first adopt patch embedding, which is implemented by a single strided 3D convolutional layer, to downsample and project each reference latent into the hidden space of the pretrained DiTs. Let $\mathcal{Y}' = \{\mathbf{y}'_s \in \mathbb{R}^{H_h \times W_h \times C_h}\}_{s=1}^{S}$ denote the projected latents. To distinguish the latents of different scales and incorporate spatial information, we construct a scale embedding $\mathbf{e}_s^{\mathrm{scale}} \in \mathbb{R}^{C_h}$ and a spatial positional embedding $\mathbf{e}^{\mathrm{pos}} \in \mathbb{R}^{H_h \times W_h \times C_h}$, and add them to $\mathbf{y}'_s$. Let $\mathbf{y}'' \in \mathbb{R}^{S \cdot H_h \cdot W_h \times C_h}$ denote the sequence-wise concatenation of all flattened $\mathbf{y}'_s$, the second cross-attention block now can be formulated as $\mathbf{q}'' = \phi(\mathbf{q}', \mathbf{y}'', \mathbf{y}'')$. Finally, we reshape $\mathbf{q}''$ to $C_h \times N \times H_h \times W_h$ and use it to enhance the reference latent in our model.

Temporal Injection and Segmented Self-Attention. As the self-attention operation is the core component of DiTs, we propose a segmented self-attention module with temporally adaptive injection to leverage the multi-scale reference latent generated by the scheduler efficiently and effectively. Particularly, we still denote the noise latent in DiTs after patch embedding as $\mathbf{x}$, and incorporate the multi-scale reference latent into it as follows:

$$\mathbf{x}' = [\mathbf{x}_{\mathrm{ref}}, \mathbf{q}''_1, \ldots, \mathbf{q}''_N, \mathbf{x}_1, \ldots, \mathbf{x}_N]. \tag{2}$$

Here, $\mathbf{x}_{\mathrm{ref}}$ and $\{\mathbf{x}_n\}_{n=1}^{N}$ are the components in $\mathbf{x}$ corresponding to the original reference image and the video latent. We further split the video latent into N segments, each of length $\frac{T_l}{N}$ and corresponding to one slot of the multi-scale reference latent. As shown in Fig. 8, we then conduct segmented self-attention following three rules: (i) the original reference latent $\mathbf{x}_{\mathrm{ref}}$ attends to all video latents $\{\mathbf{x}_n\}_{n=1}^{N}$, (ii) each slot of the multi-scale reference latent $\mathbf{q}''_n$ attends only to itself and its corresponding segment $\mathbf{x}_n$, (iii) $\mathbf{x}_n$ attends to $\mathbf{x}_{\mathrm{ref}}$, $\mathbf{q}''_n$, and all video segments. This design, especially the second rule, enables each slot to focus on its designated temporal segment without escalating the computational cost. Furthermore, with the first rule, the global information of the reference image is preserved during the generation process, while the third rule promotes temporal consistency of generated videos.

3.3 Temporal Anchoring Regularization

Conventional R2V models treat the generation process as a black-box mapping from noise to pixels and lack explicit supervision over intermediate temporal dynamics within the DiTs. This absence of process-level guidance leads to two coupled failure modes. First, without intermediate anchors, the latent trajectory from the reference condition to the output is unconstrained, making the model prone to cumulative drift in product appearance, camera pose, and inter-frame geometric consistency. Second, gradients computed solely at the output layer must traverse the full network depth to reach early blocks, where they attenuate substantially, leaving deep intermediate features under-supervised. To address these limitations, our proposed TAR leverages boundary-conditioned constraints generated by a temporal anchor branch to provide stable block-level feature supervision, enabling direct self-distillation on the internal representations.

Temporal Anchor Branch. TAR introduces the anchor branch only during training. As described in Sec. 3.1, the main branch takes the reference image, text prompt, and noisy video latent as inputs. In contrast, the anchor branch is conditioned on additional boundary information from the first and last frames of the training video, which are encoded by the VAE encoder to construct an anchored latent $\mathbf{z}^{\mathrm{anc}}$. The anchor branch is first adapted under this boundary-conditioned inpainting setting and is frozen during main branch training. Given the same noisy video latent and denoising step as the main branch, the frozen anchor branch generates boundary-aware hidden features at selected DiT blocks. Specifically, let $\mathbf{g}_l^t$ denote the hidden state from the anchor branch at denoising step t and sampled block l, and let $\mathbf{h}_l^t$ denote the corresponding hidden state from the main branch:

$$\mathbf{g}_l^t = f_{\mathrm{anc}}^l(\mathbf{x}_t, t, \mathbf{z}^{\mathrm{anc}}, \mathbf{p}), \quad \mathbf{h}_l^t = f_\theta^l(\mathbf{x}_t, t, \mathbf{z}, \mathbf{p}), \tag{3}$$

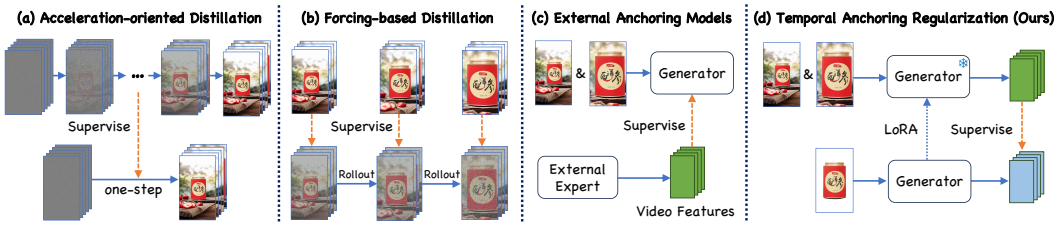

Figure 4: Comparison between TAR and other supervision paradigms. (a) Acceleration-oriented distillation compresses the multi-step denoising generation process into fewer steps. (b) Forcing-based distillation aggressively generates video frames to suppress error accumulation. (c) Anchoring models rely on external experts to provide supervision. (d) TAR not only eliminates the external dependency but also provides intermediate features for supervision.

where f^l_{anc} denotes the frozen anchor branch up to block l, and f^l_θ denotes the trainable main branch up to the same block. These anchor features provide stable temporal targets while the inference-time main branch still uses only the reference image and text prompt as external conditions, with no first- or last-frame boundary inputs.

Block-Level Feature Regularization. To balance between training efficiency and effectiveness, we select a subset $\mathcal{B}$ of DiT blocks at a fixed interval of Δl layers and collect the corresponding hidden features. We then minimize the discrepancy between the hidden features from our model and those from the anchor branch. This block-level supervision serves two purposes: First, it injects gradient signals directly into intermediate layers, mitigating the depth-induced attenuation associated with output-only losses. Second, it constrains our model's internal feature trajectories to remain consistent with the boundary-anchored representations at every sampled depth, improving temporal stability and appearance fidelity. TAR is formulated as:

$$\mathcal{L}_{\text{TAR}} = \frac{1}{|\mathcal{B}|} \sum_{l \in \mathcal{B}} w(t) \, \left\| \text{sg}(\mathbf{g}^t_l) - \mathbf{h}^t_l \right\|_2^2, \tag{4}$$

where $w(t)$ is the timestep-dependent flow-matching weight [35], which emphasizes supervision at noisier timesteps where coarse spatiotemporal structure is primarily determined, and $\text{sg}(\cdot)$ denotes stop-gradient. Since STS introduces additional reference tokens into the main branch, feature matching is applied only to the generated-video hidden tokens shared by both branches. With this regularization, the original opaque end-to-end generation process gains explicit intermediate anchors, encouraging stable feature-trajectory interpolation for intermediate frames.

Discussion. Fig. 4 highlights the differences between TAR and several related supervision paradigms in video generation. Compared with acceleration-oriented [36, 46, 55, 58, 59] or forcing-based distillation methods [7, 25, 28], our method does not aim to compress or imitate a teacher denoising trajectory, but instead regularizes the internal feature evolution of the generator under the original generation process. Compared with external anchoring approaches, TAR avoids the representation gap introduced by heterogeneous expert models [4, 61] or the dependency of first and last frames for inference [50, 57], since its supervision is produced by its frozen counterpart. By constructing boundary-conditioned anchor features from the first and last frames, TAR provides explicit intermediate guidance for temporally stable feature supervision, which is absent from prior distillation or external alignment paradigms.

3.4 Joint Training Objectives

The total training loss combines the standard flow-matching regression target with the TAR:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{gt}} \cdot \mathcal{L}_{\text{gt}} + \lambda_{\text{TAR}} \cdot \mathcal{L}_{\text{TAR}}, \tag{5}$$

where $\mathcal{L}_{\text{gt}} = \|f_\theta(\mathbf{x}_t, t) - \mathbf{v}^*\|_2^2 \cdot w(t)$ is the loss on the target velocity field $\mathbf{v}^*$ [35], θ denotes the learnable parameters in STS modulation, and λ_{gt} and λ_{TAR} balance the two objectives.

Table 1: Quantitative comparison with existing methods. AdDirector achieves markedly superior performance in subject consistency and camera accuracy. Best values are **bold**, second-best are underlined. All metrics are higher-is-better (↑).

Model	Camera Acc.			Subject Cons.			Video Qual.		
	Cam.IS↑	VQAScore↑	Human↑	CLIP-I↑	DINO-I↑	Human↑	Aes.Qual.↑	Motion↑	Human↑
Vace 14B [27]	0.410	0.3700	17.0%	0.9334	0.9022	21.6%	0.4994	0.9875	21.2%
Phantom 14B [37]	0.310	0.2835	2.8%	0.9257	0.8817	6.1%	0.4810	0.9937	4.4%
SkyReels-A2 [16]	0.315	0.3500	3.4%	0.9384	**0.9085**	6.5%	0.4804	0.9926	4.9%
HunyuanCustom [23]	0.405	0.3850	3.1%	0.9084	0.8514	4.0%	0.4636	0.9886	5.3%
MAGREF [14]	0.325	0.3800	4.9%	0.8982	0.8188	2.3%	**0.5200**	0.9903	6.0%
BindWeave [33]	0.280	0.3400	2.9%	0.9062	0.8617	2.3%	0.4369	0.9816	1.9%
Pika2.5 [41]	0.380	0.2900	3.4%	0.9352	0.8908	4.5%	0.4662	0.9934	7.0%
Keling1.6 [29]	0.525	0.3450	6.8%	0.9319	0.8860	5.3%	0.4753	**0.9954**	8.9%
AdDirector (Ours)	**0.840**	**0.4750**	**55.5%**	**0.9412**	0.9006	**47.4%**	0.4423	0.9943	**40.3%**

4 Experiments

4.1 Datasets, Metrics and Implementation Details

Datasets. In the absence of any publicly available dataset tailored to product advertisement videos with controllable camera motion, we curate a dedicated dataset comprising approximately 45K samples sourced from both real-world recordings and synthetically generated content. The real-world subset is collected from mainstream video platforms and consists of human-captured product demonstrations. The synthetic subset is generated either by proprietary image-to-video generation models or through Adobe After Effects (AE) animation pipelines. The real-to-synthetic ratio is approximately 3:1, ensuring both realism and camera motion diversity. The captions are generated with Qwen-VL [48], while camera instructions are meticulously annotated by human experts to ensure precision. For evaluation, we assemble a 200-sample benchmark, of which 100 are from independent real-world videos not included in the training set, and 100 are carefully crafted high-fidelity brand advertisement prompts. Camera types are evenly distributed to ensure a balanced assessment.

Metrics. We evaluate our camera-controllable R2V generation framework using both automatic metrics and human evaluation across three aspects: camera accuracy, subject consistency, and video quality. For camera accuracy, we adopt VQAScore from CameraBench [34], which uses a fine-tuned VLM to classify camera categories in generated videos. We additionally report the Camera Instruction Score (Cam. IS) [39], a binary classification metric that assesses whether the generated video correctly follows the specified camera instruction. To measure subject consistency, we uniformly sample 10 frames per video and compute the similarity with the reference image using CLIP [12] and DINO [40] features. For video quality, we adopt two sub-dimensions from VBench [26], namely aesthetic quality and motion smoothness, to evaluate perceptual fidelity and temporal coherence. The Human column in Tab. 1 reports aggregated preference ratios from our user study.

Implementation Details. Our AdDirector framework is built upon the Wan architecture [11] and consists of two branches: a main branch and an anchor branch. Training of the main branch is conducted on 32 NVIDIA H20 GPUs with a batch size of 1. LoRA parameters are fine-tuned for 1,300 steps on videos of resolution 832×480 and 81 frames per video, employing the Adam optimizer with a learning rate of 1×10^{-4}. The anchor branch is trained separately on 16 NVIDIA H20 GPUs, also with a batch size of 1, with LoRA fine-tuned for 10,000 steps. For TAR, the main branch is updated while the anchor branch parameters remain frozen, and the hyperparameters are set as $\lambda_{\text{gt}} = 0.5$, $\lambda_{\text{TAR}} = 0.5$, and $\Delta l = 8$. For STS, we set the number of multi-scale candidates to $S = 9$ and the number of temporal slots to $N = 3$. We perform 40 denoising steps and adopt a classifier-free guidance scale of 5 to generate the final videos for inference.

4.2 Quantitative Results

We quantitatively compare our proposed method against recent state-of-the-art reference-to-video generation models, including the closed-source models Pika2.5 [41], and Keling1.6 [29], as well as the open-source models Vace [27], Phantom [37], SkyReels-A2 [16], HunyuanCustom [23], MAGREF [14], and BindWeave [33], as summarized in Tab. 1. Our AdDirector exhibits outstand-

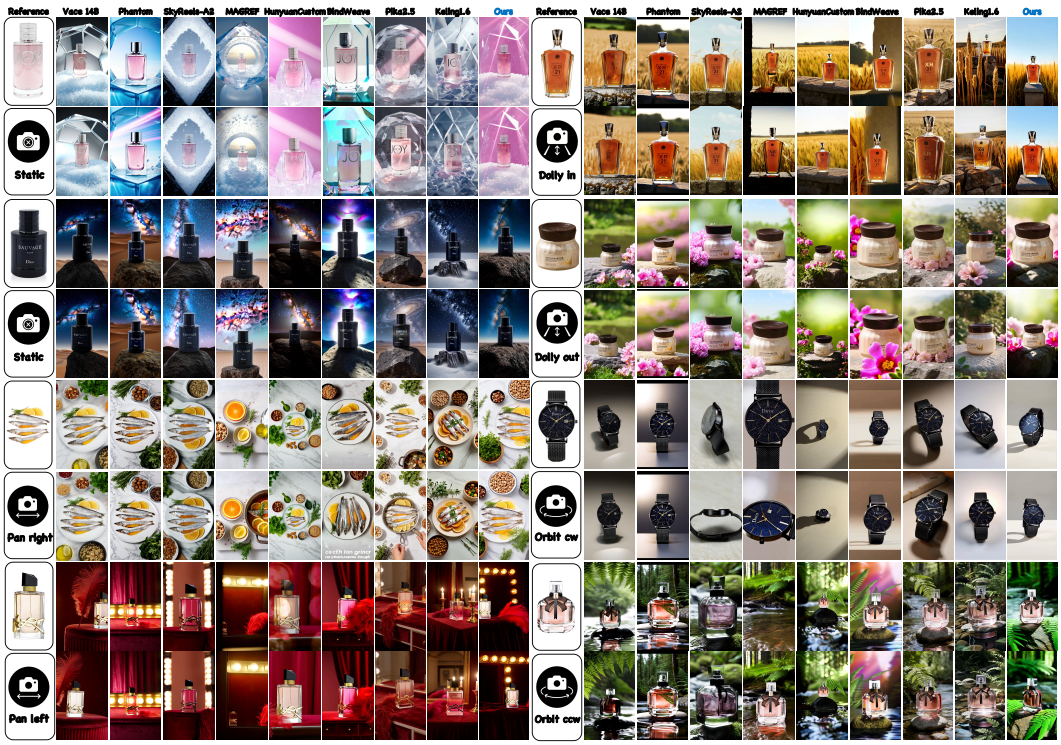

Figure 5: Qualitative results with existing methods. AdDirector is capable of producing accurate camera motion driven purely by semantic instructions, while maintaining the clarity and spatial coherence of key visual elements (e.g., logos) across camera motion. Best viewed on screen.

ing subject consistency, faithfully preserving both the identity and fine-grained appearance of the reference image. More strikingly, in terms of camera control, AdDirector achieves an accuracy of 84%, demonstrating that high-precision video camera motions can be reliably generated solely from semantic camera instructions. We further assess performance across diverse camera types, including static, dolly, orbit/rotate, and pan shots. As illustrated in Fig. 9, AdDirector maintains consistently high accuracy across all categories, underscoring its robustness and precision in camera control. Collectively, these results substantiate that AdDirector can generate videos with both highly controllable camera motion and superior subject fidelity.

4.3 Qualitative Results

The qualitative results of AdDirector compared with representative R2V models are shown in Fig. 5. The visual results reveal that existing methods [14, 16, 23, 27, 29, 33, 37, 41] exhibit limited controllability when executing semantic camera instructions. Specifically, Phantom [37] and SkyReels-A2 [16] tend to produce overly static outputs and show weak responsiveness to dynamic motions such as dolly and rotation, whereas Pika2.5 [41] and Keling1.6 [29] display a pronounced bias toward dynamic motion, often compromising the faithful rendering of static shots. In comparison, AdDirector consistently adheres to a broad spectrum of camera instructions, producing high-quality videos with accurate and stable motion patterns. Furthermore, existing methods also fail to maintain strict cross-frame appearance alignment, leading to artifacts such as logo blurring or drifting. AdDirector, by contrast, maintains robust appearance consistency throughout camera movement, preserving the clarity of critical visual elements and preventing geometric deformation across frames. Collectively, these qualitative results substantiate the superiority of AdDirector in achieving both precise camera control and rigorous appearance consistency.

4.4 Ablation Study

To validate the effectiveness of the AdDirector framework, we conduct a systematic ablation study on its two core components, i.e., STS and TAR. The results (the first row in Tab. 2) demonstrate that

Table 2: Ablation studies on the key components in AdDirector.

Module		Camera Acc.		Subject Cons.		Video Qual.		Computation Cost		
STS	TAR	Cam.IS↑	VQAScore↑	CLIP-I↑	DINO-I↑	Aes.Qual.↑	Motion↑	Param.(B)	Mem.(G)	Time(min)
×	×	0.485	0.3700	0.9248	0.8781	0.4283	0.9924	16.70	39.49	25.53
✓	×	0.695	<u>0.4500</u>	<u>0.9306</u>	<u>0.8964</u>	<u>0.4376</u>	0.9934	17.56	44.78	28.03
×	✓	<u>0.740</u>	0.4450	0.9304	0.8872	0.4275	<u>0.9938</u>	16.70	39.49	25.53
✓	✓	**0.840**	**0.4750**	**0.9412**	**0.9006**	**0.4423**	**0.9943**	17.56	44.78	28.03



Figure 6: Ablation studies of AdDirector. STS and TAR exhibit complementary and synergistic effects. Their joint integration produces superior results with improved fidelity and camera accuracy.

fine-tuning the base model on paired data alone achieves less than 50% camera accuracy, indicating that data adaptation is insufficient to reliably align semantic camera instructions with actual motion. Fig. 6 illustrates that the absence of STS during a dolly shot leads to blurred product logos and reduced camera controllability, whereas omitting TAR results in unstable camera motion and pronounced feature drift. Tab. 2 further indicates that STS substantially enhances subject consistency, while TAR significantly improves camera control. When applied jointly, the model achieves peak performance across all metrics, highlighting the complementary and synergistic effects. The results demonstrate that STS, through multi-scale reference scheduling, significantly enhances subject consistency, effectively preserving the clarity and coherence of key visual elements (e.g., logos) during camera motion while substantially reducing blur artifacts. TAR, by providing anchored temporal features to refine the intermediate representations within the DiT blocks, not only produces smoother video outputs but also substantially mitigates feature drift. Both STS and TAR contribute significantly to improved camera control. When applied jointly, the model achieves peak performance across all metrics, highlighting the complementary and synergistic effects of these two modules.

4.5 User Study

We conducted a user study with 30 participants to evaluate the results. For each participant, 20 videos were randomly sampled, covering all camera motion types proportionally. For each video, participants selected the best-generated result along three dimensions: camera accuracy, video quality, and subject consistency. The aggregated preference ratio for each dimension was then computed and reported as the human evaluation score. As shown in Tab. 1, the results indicate that our method attains the highest preference ratios across all three dimensions.

5 Conclusion

In this paper, we propose AdDirector, a novel reference-to-video framework for camera-controllable advertisement video generation. AdDirector introduces spatio-temporal-scale modulation to adaptively inject multi-scale reference features through camera-conditioned 3D tokens, enabling balanced preservation of global structure and fine-grained details. It further incorporates temporal anchoring regularization, which employs a frozen first-and-last-frame anchor branch to stabilize intermediate features and reduce temporal drift. Extensive experiments demonstrate that AdDirector achieves superior camera controllability and product consistency without requiring auxiliary control signals.

Acknowledgements

This work is supported by National Key Research and Development Program of China (2024YFE 0203100), Scientific Research Innovation Capability Support Project for Young Faculty (No.ZYGXQNJSKYCXNLZCXM-I28), National Natural Science Foundation of China (NSFC) under Grants No.62476293, Shenzhen Science and Technology Program (Grant No. QNXMA20250701093544048) and General Embodied AI Center of Sun Yat-sen University. This work was supported in part by The Major Key Project of PCL (Grant No. PCL2025A17).

References

[1] Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024.

[2] Sherwin Bahmani, Tianchang Shen, Jiawei Ren, Jiahui Huang, Yifeng Jiang, Haithem Turki, Andrea Tagliasacchi, David B. Lindell, Zan Gojcic, Sanja Fidler, Huan Ling, Jun Gao, and Xuanchi Ren. Lyra: Generative 3d scene reconstruction via video diffusion model self-distillation. In *International Conference on Learning Representations*, 2026.

[3] Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, et al. Recammaster: Camera-controlled generative rendering from a single video. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14834–14844. IEEE, 2025.

[4] Aritra Bhowmik, Denis Korzhenkov, Cees GM Snoek, Amirhossein Habibian, and Mohsen Ghafoorian. Moalign: Motion-centric representation alignment for video diffusion models. *arXiv preprint arXiv:2510.19022*, 2025.

[5] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[6] Yiyang Cai, Nan Chen, Rongchang Xie, Junwen Pan, Chunyang Jiang, Cheng Chen, Wen Zhou, Zhenbang Sun, Wei Xue, Wenhan Luo, et al. Homie: Human-object centric video personalization via multimodal intelligent enchancement. *arXiv preprint arXiv:2607.18217*, 2026.

[7] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

[8] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023.

[9] Nan Chen, Yiyang Cai, Rongchang Xie, Junwen Pan, Cheng Chen, Weinan Jia, Zhuowei Chen, Wen Zhou, Zhenbang Sun, and Wenhan Luo. Domainshuttle: Freeform open domain subject-driven text-to-video generation. *arXiv preprint arXiv:2606.26058*, 2026.

[10] Yingjie Chen, Yifang Men, Yuan Yao, Miaomiao Cui, and Liefeng Bo. Perception-as-control: Fine-grained controllable image animation with 3d-aware motion representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14380–14389, 2025.

[11] Gang Cheng, Xin Gao, Li Hu, Siqi Hu, Mingyang Huang, Chaonan Ji, Ju Li, Dechao Meng, Jinwei Qi, Penchong Qiao, et al. Wan-animate: Unified character animation and replacement with holistic replication. *arXiv preprint arXiv:2509.14055*, 2025.

[12] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2818–2829, 2023.

[13] Hyung Won Chung, Noah Constant, Xavier Garcia, Adam Roberts, Yi Tay, Sharan Narang, and Orhan Firat. Unimax: Fairer and more effective language sampling for large-scale multilingual pretraining. *arXiv preprint arXiv:2304.09151*, 2023.

[14] Yufan Deng, Xun Guo, Yuanyang Yin, Jacob Zhiyuan Fang, Yiding Yang, Yizhi Wang, Shenghai Yuan, Angtian Wang, Bo Liu, Haibin Huang, et al. Magref: Masked guidance for any-reference video generation. *arXiv e-prints*, pages arXiv–2505, 2025.

[15] Yang Fei, George Stoica, Jingyuan Liu, Qifeng Chen, Ranjay Krishna, Xiaojuan Wang, and Benlin Liu. Structure from tracking: Distilling structure-preserving motion for video generation. *arXiv preprint arXiv:2512.11792*, 2025.

[16] Zhengcong Fei, Debang Li, Di Qiu, Jiahua Wang, Yikun Dou, Rui Wang, Jingtao Xu, Mingyuan Fan, Guibin Chen, Yang Li, et al. Skyreels-a2: Compose anything in video diffusion transformers. *arXiv preprint arXiv:2504.02436*, 2025.

[17] Daniel Geng, Charles Herrmann, Junhwa Hur, Forrester Cole, Serena Zhang, Tobias Pfaff, Tatiana Lopez-Guevara, Yusuf Aytar, Michael Rubinstein, Chen Sun, et al. Motion prompting: Controlling video generation with motion trajectories. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1–12, 2025.

[18] Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

[19] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024.

[20] Hao He, Ceyuan Yang, Shanchuan Lin, Yinghao Xu, Meng Wei, Liangke Gui, Qi Zhao, Gordon Wetzstein, Lu Jiang, and Hongsheng Li. Cameractrl ii: Dynamic scene exploration via camera-controlled video diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13416–13426, 2025.

[21] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

[22] Chen Hou and Zhibo Chen. Training-free camera control for video generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[23] Teng Hu, Zhentao Yu, Zhengguang Zhou, Sen Liang, Yuan Zhou, Qin Lin, and Qinglin Lu. Hunyuancustom: A multimodal-driven architecture for customized video generation. *arXiv preprint arXiv:2505.04512*, 2025.

[24] Teng Hu, Zhentao Yu, Zhengguang Zhou, Jiangning Zhang, Yuan Zhou, Qinglin Lu, and Ran Yi. Polyvivid: Vivid multi-subject video generation with cross-modal interaction and enhancement. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[25] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025.

[26] Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, et al. Vbench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[27] Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17191–17202, 2025.

[28] Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024.

[29] Kling Team. Kling1.6 elements to video. `https://app.klingai.com/global/image-to-video/multi-id/new`, 2025.

[30] 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.

[31] Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. REPA-E: Unlocking VAE for end-to-end tuning of latent diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18262–18272, 2025.

[32] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, pages 19730–19742, 2023.

[33] Zhaoyang Li, Dongjun Qian, Kai Su, Qishuai Diao, Xiangyang Xia, Chang Liu, Wenfei Yang, Tianzhu Zhang, and Zehuan Yuan. Bindweave: Subject-consistent video generation via cross-modal integration. *arXiv preprint arXiv:2510.00438*, 2025.

[34] Zhiqiu Lin, Siyuan Cen, Daniel Jiang, Jay Karhade, Hewei Wang, Chancharik Mitra, Tiffany Ling, Yuhan Huang, Sifan Liu, Mingyu Chen, et al. Towards understanding camera motions in any video. *arXiv preprint arXiv:2504.15376*, 2025.

[35] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023.

[36] Jinxiu Liu, Xuanming Liu, Kangfu Mei, Yandong Wen, Ming-Hsuan Yang, and Weiyang Liu. Streaming autoregressive video generation via diagonal distillation. In *International Conference on Learning Representations*, 2026.

[37] Lijie Liu, Tianxiang Ma, Bingchuan Li, Zhuowei Chen, Jiawei Liu, Gen Li, Siyu Zhou, Qian He, and Xinglong Wu. Phantom: Subject-consistent video generation via cross-modal alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14951–14961, 2025.

[38] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024.

[39] Xiaofeng Mao, Zhen Li, Chuanhao Li, Xiaojie Xu, Kaining Ying, Tong He, Jiangmiao Pang, Yu Qiao, and Kaipeng Zhang. Yume-1.5: A text-controlled interactive world generation model. *arXiv preprint arXiv:2512.22096*, 2025.

[40] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[41] Pika Team. Pika2.1 consistent character video. `https://pika.art`, 2025.

[42] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[43] Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6121–6132. IEEE, 2025.

[44] Team Seedream, Yunpeng Chen, Yu Gao, Lixue Gong, Meng Guo, Qiushan Guo, Zhiyao Guo, Xiaoxia Hou, Weilin Huang, Yixuan Huang, et al. Seedream 4.0: Toward next-generation multimodal image generation. *arXiv preprint arXiv:2509.20427*, 2025.

[45] Joonghyuk Shin, Zhengqi Li, Richard Zhang, Jun-Yan Zhu, Jaesik Park, Eli Shechtman, and Xun Huang. Motionstream: Real-time video generation with interactive motion controls, 2025. *URL https://arxiv. org/abs/2511.01266*, 2025.

[46] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, pages 32211–32252, 2023.

[47] Tomás Soucek and Jakub Lokoc. Transnet v2: An effective deep network architecture for fast shot transition detection. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 11218–11221, 2024.

[48] Qwen Team. Qwen2.5-vl, January 2025. URL `https://qwenlm.github.io/blog/qwen2.5-vl/`.

[49] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[50] Xiaojuan Wang, Boyang Zhou, Brian Curless, Ira Kemelmacher-Shlizerman, Aleksander Holynski, and Steven M. Seitz. Generative inbetweening: Adapting image-to-video models for keyframe interpolation. In *International Conference on Learning Representations*, 2025.

[51] Zhaoqing Wang, Xiaobo Xia, Zhuolin Bie, Jinlin Liu, Dongdong Yu, Jia-Wang Bian, and Changhu Wang. Taming camera-controlled video generation with verifiable geometry reward. *arXiv preprint arXiv:2512.02870*, 2025.

[52] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

[53] Shiyuan Yang, Liang Hou, Haibin Huang, Chongyang Ma, Pengfei Wan, Di Zhang, Xiaodong Chen, and Jing Liao. Direct-a-video: Customized video generation with user-directed camera movement and object motion. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–12, 2024.

[54] Xiaoda Yang, Jiayang Xu, Kaixuan Luan, Xinyu Zhan, Hongshun Qiu, Shijun Shi, Hao Li, Shuai Yang, Li Zhang, Checheng Yu, et al. Omnicam: Unified multimodal video generation via camera control. *arXiv preprint arXiv:2504.02312*, 2025.

[55] Yongqi Yang, Huayang Huang, Xu Peng, Xiaobin Hu, Donghao Luo, Jiangning Zhang, Chengjie Wang, and Yu Wu. Towards one-step causal video generation via adversarial self-distillation. In *International Conference on Learning Representations*, 2026.

[56] 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. In *The Thirteenth International Conference on Learning Representations*, 2025.

[57] Zuhao Yang, Jiahui Zhang, Yingchen Yu, Shijian Lu, and Song Bai. Versatile transition generation with image-to-video diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16981–16990, 2025.

[58] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6613–6623, 2024.

[59] Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22963–22974, 2025.

[60] Mark Yu, Wenbo Hu, Jinbo Xing, and Ying Shan. Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 100–111. IEEE, 2025.

[61] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

[62] Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

[63] Zijian Zhou, Shikun Liu, Haozhe Liu, Haonan Qiu, Zhaochong An, Weiming Ren, Zhiheng Liu, Xiaoke Huang, Kam Woh Ng, Tian Xie, et al. Scaling zero-shot reference-to-video generation. *arXiv preprint arXiv:2512.06905*, 2025.

A Introduction of Supplementary Material

This supplementary material provides additional method details and experimental results to complement the main paper. Specifically, Sec. B describes the dataset construction and annotation pipeline, including data sources, video processing, text and camera-motion annotation, and reference-image extraction. Sec. C provides further analysis of STS, including attention visualizations for adaptive scale selection and zero-shot complex camera-motion modeling. Sec. D presents additional training and evaluation details, quantitative comparisons with explicit camera-control methods, additional aesthetic evaluation, qualitative results, hyperparameter ablations, and user studies. Finally, Sec. E discusses the limitations of AdDirector, while Sec. F addresses its broader social impact and potential risks. We strongly recommend exploring the accompanying **HTML file** for intuitive video visualizations highlighting the camera controllability and appearance fidelity of AdDirector.

B Dataset Construction and Annotation

To support advertisement video generation, we construct a dedicated dataset containing approximately **45K samples**. The dataset is composed of both **real-world captured videos** and **synthetic video content**, enabling a balance between visual realism and diverse camera motion patterns.

- **Data Sources.** The dataset contains two types of video sources:
  - **Real-world videos** collected from mainstream video platforms, consisting of product showcase videos captured by human photographers. These videos provide realistic presentation scenarios with natural lighting, scene composition, and camera movements.
  - **Synthetic videos** constructed from two sources: (1) animated advertising materials created using Adobe After Effects (AE), and (2) videos generated by a proprietary image-to-video generation model. All synthetic samples are standardized in format and further filtered through quality screening to ensure consistency with the real-world data.

  The overall ratio between real and synthetic data is approximately **3:1**, which preserves realism while enriching the diversity of camera motion patterns.
- **Dataset Structure.** Each sample in the dataset contains four components:
  - a **product reference image**,
  - a **text description**,
  - a **camera motion annotation**,
  - and a **product advertisement video clip**.
- **Video Processing.** For the video component, real-world videos are first processed using TransNet v2 [47] to perform initial shot boundary detection. We further apply a dynamic threshold refinement strategy: long shots are subdivided to obtain finer temporal segments, while extremely short shots are merged with their subsequent shots to maintain temporal coherence. After segmentation, we employ VLM [48] to perform rule-based preliminary filtering of individual shots (Tab. 3). The resulting clips are then manually inspected to remove low-quality segments and non-advertisement content, yielding a clean set of product advertisement video clips.
- **Text Annotation.** We utilize Qwen-VL [48] to automatically generate textual descriptions for each video clip, establishing a semantic bridge between the visual content and textual descriptions. This process provides consistent multimodal supervision for model training.
- **Camera Motion Annotation.** Camera motion labels are annotated manually. Since accurate recognition of camera movements requires high precision and existing vision-language models or pose estimation methods exhibit limited reliability for this task, human annotation is adopted to ensure accurate and unambiguous camera motion labels. The camera motion annotations encompass four categories—static, dolly, orbit/rotate, and pan—with their respective proportions in the dataset estimated at approximately 53%, 24%, 14%, and 9%, thereby reflecting a comprehensive coverage of both stable and dynamic cinematographic scenarios.

Table 3: Rule-based filtering criteria for advertisement video clips.

Criterion	Description
Subject clarity	The product appears clearly with a prominent subject and stable visual focus.
Filming artifacts	No filming equipment, camera operators, or hands manipulating the product or adjusting the camera.
Watermarks & subtitles	Brand logos, promotional information, or stylized advertisement subtitles are allowed. However, filming-related overlays such as shooting parameters, device status, operational prompts, countdown timers, or recording indicators are not permitted.
Visual quality	The clip should not contain large mosaic artifacts, black frames, flickering, or obvious traces of on-site lighting adjustment.
Content validity	The clip should not be a recording of screen playback (e.g., where the main subject is a screen or projection), nor a static presentation of still images.

Table 4: Guidelines for the user study. For each dimension, participants are asked to select the best video among the compared methods.

Criterion	Evaluation Guidelines
Camera accuracy	Select the video whose camera motion best follows the specified instruction (e.g., pan, dolly, orbit), considering whether the motion direction, trajectory, and pattern match the intended camera movement.
Video quality	Select the video with the best overall perceptual quality, considering visual aesthetics, motion smoothness, temporal stability, and the absence of artifacts such as flickering, distortion, or abrupt transitions.
Subject consistency	Select the video in which the generated subject is most consistent with the reference product image, focusing on attributes such as logo appearance, shape, structure, color, and distinctive visual details.

- **Reference Image Extraction.** Product reference images are extracted directly from the videos. We employ GroundingDINO [38] and SAM2 [42] to segment the primary product from video frames. The segmentation results are then filtered to retain only high-quality images where the product is clearly visible, structurally complete, and rich in visual details. These images serve as reference inputs for generating advertisement videos.

C Additional Method Details

Adaptive Scale Selection in STS. Fig. 8 provides a detailed illustration of the segmented self-attention design. Segmented self-attention allows each reference slot to focus on its designated temporal segment, retain global context, and maintain temporal coherence while avoiding excessive computational cost. Fig. 7 visualizes the candidate-level attention distributions across different camera motions and temporal slots. The heatmaps show that STS dynamically shifts its attention among global, cropped, and downsampled references rather than consistently favoring a single scale:

- **Dolly.** As the shot pushes in, attention on downsampled candidates is high early and gradually decays, while attention on high-resolution cropped candidates progressively rises, consistent with the subject growing larger and local details becoming more important.
- **Static.** Attention stays high on both downsampled and cropped candidates, indicating that the global contour and local high-resolution details are used complementarily rather than being dominated by a single scale.
- **Orbit/Rotate.** Attention favors the more macroscopic global and downsampled references, with consistently weak and nearly flat attention on cropped candidates, showing reliance on overall scene structure and global object appearance rather than local regions.
- **Pan.** Attention on crop is higher in the first and last segments but shifts toward downsample in the middle. This aligns with the subject's changing visible extent: the boundary segments show only local regions (favoring high-resolution local references), while the middle presents a more complete subject (favoring low-resolution references that capture the overall contour).

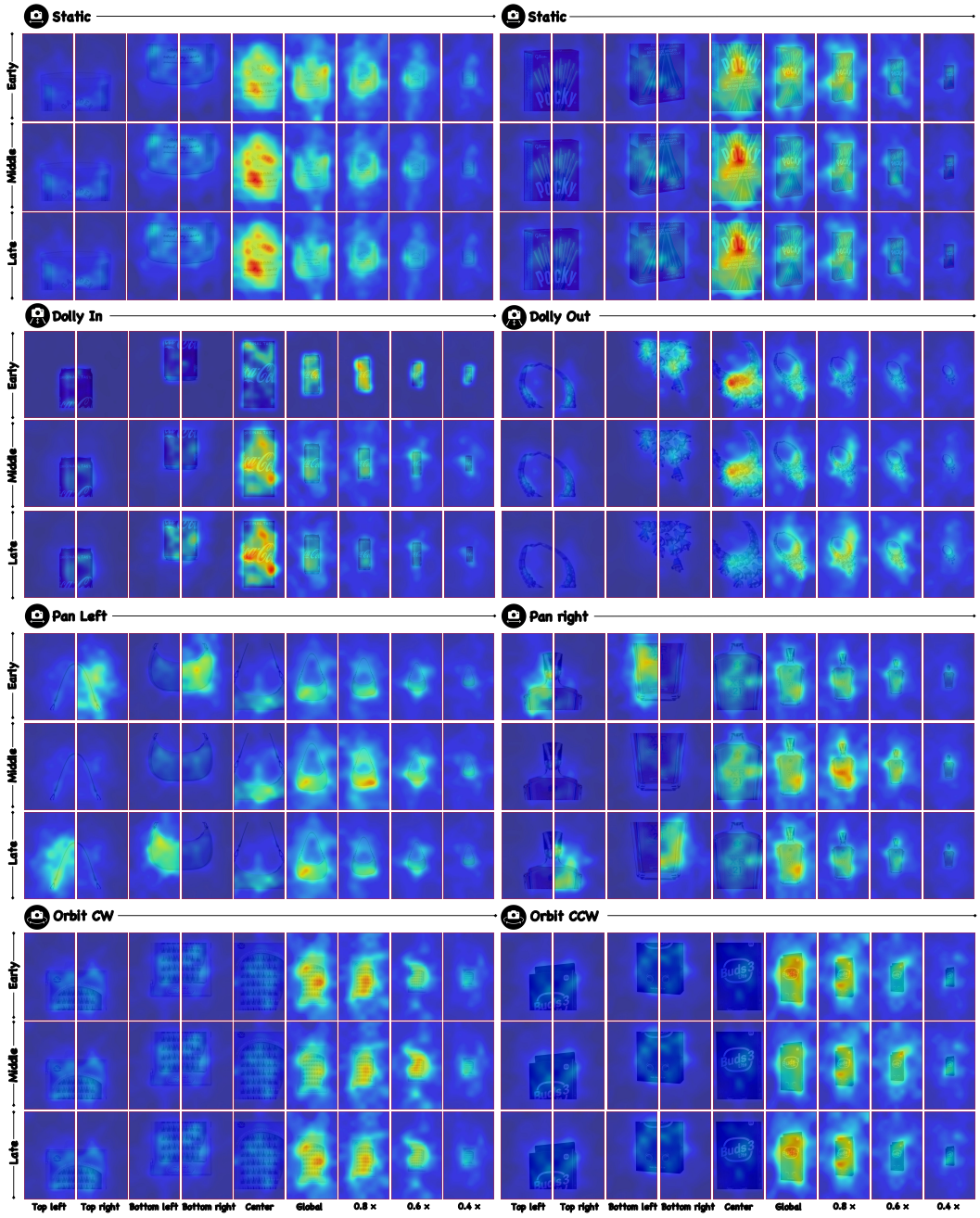

Figure 7: Candidate-level attention distributions of STS across different camera motions and temporal slots. The attention patterns dynamically vary among global, cropped, and downsampled reference candidates, demonstrating adaptive scale selection.

Zero-shot Complex Camera Motions Modeling. For complex camera motions, text prompts do not directly regress continuous explicit camera trajectories. Instead, they are first encoded by the T5 encoder into text tokens that contain camera-motion semantics. A complex camera motion can usually be represented as a composition of multiple basic camera-motion semantics, such as dolly and pan. Accordingly, for each basic camera-motion semantic involved in the text prompt, STS uses the corresponding camera CLS embedding to cross-attend with the T5 text tokens in the Text Layer, thereby extracting the semantic intent of that basic camera motion from the text prompt. Then, in the Visual Layer, the Reference Scheduler selects and aggregates relevant reference queries

from multi-scale reference features according to this semantic intent, producing the corresponding camera-conditioned reference features. When the text prompt contains multiple basic camera-motion semantics, these camera-conditioned reference features are fused at the feature level and used as the combined condition for the Main branch. Therefore, text prompts model complex camera motions through T5 semantic encoding, camera CLS conditioning, and STS reference scheduling, rather than relying on explicit trajectory annotations or continuous camera-parameter regression.

D Additional Experimental Results

D.1 Additional Implementation Details

Additional Training Details. During training, we set both the LoRA rank and alpha to 64. We adopt a text dropout strategy with a probability of 20%, wherein the original prompt—comprising both a visual description and a camera instruction—is replaced by a simplified prompt retaining only the camera instruction. To mitigate the inherent class imbalance among camera motion types in the training data, we additionally employ a weighted sampler to ensure balanced sampling. These configurations are applied consistently across all ablation experiments to ensure fair comparisons.

Evaluation Set Composition. Our evaluation set contains 200 samples and is constructed to ensure balanced and diverse coverage of camera motions. It consists of two subsets. The first subset contains 100 unseen samples that are held out from training and balanced in a 1:1:1:1 ratio across static, pan, orbit, and dolly camera motions. The second subset contains 100 curated reference–text pairs. Specifically, we select 20 distinct products and construct five camera-motion instructions for each product, covering different camera types and directions. This design enables each product to be evaluated under multiple camera motions rather than a single predefined condition.

Multi-Scale Candidate Construction. Given a reference product image, we build a total of $S = 9$ candidates spanning global, downsampled, and cropped scales: (1) 1 global candidate: the original product reference image at its native scale; (2) 3 downsampled candidates: obtained by downsampling the centered product region at ratios of 0.8, 0.6, and 0.4; (3) 5 cropped candidates: four quadrant crops (top-left, top-right, bottom-left, and bottom-right) and one high-resolution center crop.

Temporal Positional Encoding. For the multi-scale reference latents introduced in the proposed STS module, we assign virtual temporal positional encodings that are deliberately decoupled from those of the video frames. This separation prevents the model from misidentifying the reference latents as initial frames, thereby avoiding contamination of the first-frame generation process.

D.2 More Quantitative Results

Performance across Camera Types. We assess performance across diverse camera types, including static, dolly, orbit/rotate, and pan shots. As illustrated in Fig. 9, AdDirector maintains consistently high accuracy across all categories, underscoring its robustness and precision in camera control. On the curated reference–text subset, AdDirector achieves a macro-averaged camera accuracy of 0.80, computed by first evaluating the accuracy for each product and then averaging over all 20 products.

Comparison with Explicit Camera-control Methods. We compare AdDirector with explicit camera-control methods under two complementary settings. **(1) Direct video-to-video explicit-control comparison.** We first compare with ReCamMaster [3] and TrajectoryCrafter [60], two representative video-to-video methods conditioned on explicit target camera trajectories. Unlike these methods, AdDirector generates videos from a single product reference image and a semantic camera instruction without requiring an explicit trajectory. To adapt them to our setting, we construct a static product video by repeating the reference frame and use it as their visual input. We evaluate all methods on 101 samples uniformly selected from the non-static camera categories of the unseen test subset. As shown in Tab. 5, AdDirector achieves comparable camera-control accuracy while substantially outperforming these methods in product fidelity, as reflected by its higher CLIP-I and DINO-I scores. **(2) Cascaded image-editing and explicit-control comparison.** We further benchmark two-stage pipelines that first synthesize an edited product view and then apply an explicit camera-control method. We construct a 60-sample benchmark by uniformly sampling

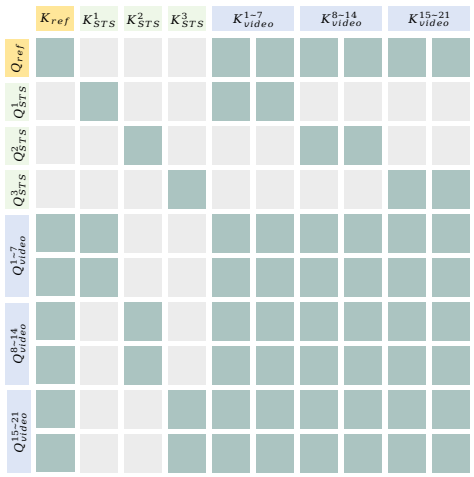

Figure 8: Segmented self-attention (SSA). It allows each reference slot to focus on its designated segment, retain global context, and maintain temporal coherence, while avoiding excessive computational cost.

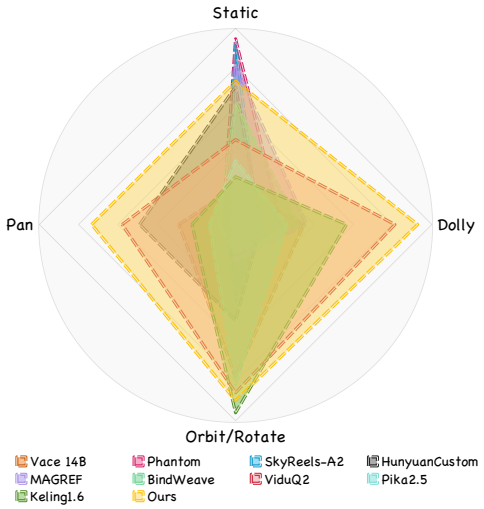

Figure 9: The accuracy across different camera types. It demonstrates that AdDirector consistently maintains high performance, highlighting its robustness and precision in camera motion control.

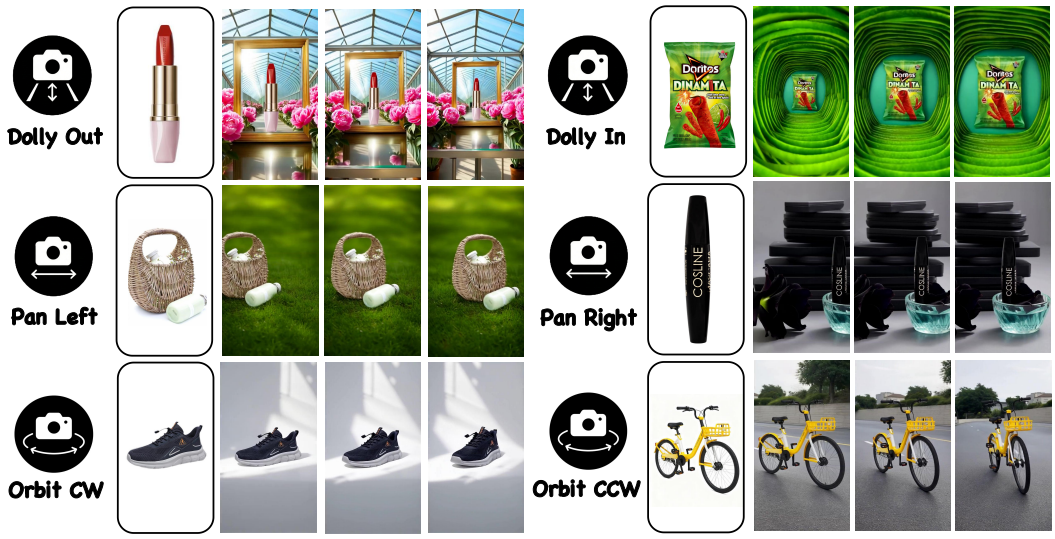

Figure 10: Additional visual results for diverse camera directions, including dolly in/out, pan left/right, and clockwise/counterclockwise orbit.

different camera types and motion directions. For image editing, we use Qwen-Image-Edit [52] and Seedream 4.0 [44]; for explicit control, we evaluate ReCamMaster [3], TrajectoryCrafter [60], GEN3C [43], and Perception-as-Control [10], resulting in eight pipeline variants. As reported in Tab. 6, these pipelines do not simultaneously achieve strong camera adherence and product fidelity. AdDirector achieves a competitive Cam.IS of 0.8833 and the highest CLIP-I and DINO-I scores of 0.9571 and 0.9349, respectively. Although Qwen-Image-Edit combined with ReCamMaster achieves a higher Cam.IS using frame-wise camera extrinsics, it exhibits lower product fidelity. **(3) Data and deployment cost.** The difference between AdDirector and explicit geometric camera-control methods lies not only in the form of the control signal, but also in the cost of data construction and practical usage. ReCamMaster [3] requires multi-camera synchronized videos with corresponding camera trajectories, and its public dataset is constructed from multi-camera videos rendered in Unreal

Table 5: Comparison with explicit geometric camera-control methods. Unlike the competing methods, AdDirector requires neither an explicit target camera trajectory nor geometric camera conditions.

	Camera Acc.		Subject Cons.		Video Qual.	
Model	**Cam.IS↑**	**VQAScore↑**	**CLIP-I↑**	**DINO-I↑**	**Aes.Qual.↑**	**Motion↑**
ReCamMaster [3]	0.8812	0.5248	0.8794	0.8205	0.4078	0.9960
TrajectoryCrafter [60]	0.8020	0.4950	0.8912	0.8337	0.4181	0.9957
AdDirector (Ours)	0.8614	0.5050	0.9411	0.9125	0.4123	0.9951

Table 6: Comparison with cascaded image-editing and explicit camera-control pipelines on a 60-sample benchmark.

	Camera Acc.		Subject Cons.		Video Qual.	
Method	**Cam.IS↑**	**VQAScore↑**	**CLIP-I↑**	**DINO-I↑**	**Aesthetic↑**	**Smoothness↑**
Qwen-Image-Edit + ReCamMaster	0.9167	0.5667	0.9349	0.9103	0.4813	0.9958
Qwen-Image-Edit + TrajectoryCrafter	0.8000	0.3333	0.9430	0.9163	0.5097	0.9956
Qwen-Image-Edit + GEN3C	0.8500	0.5667	0.8929	0.8501	0.4485	0.9887
Qwen-Image-Edit + Perception-as-Control	0.5833	0.2833	0.9332	0.9145	0.4821	0.9854
SeedDream 4.0 + ReCamMaster	0.8333	0.6000	0.9167	0.8786	0.4735	0.9954
SeedDream 4.0 + TrajectoryCrafter	0.8667	0.4000	0.9256	0.8910	0.5035	0.9952
SeedDream 4.0 + GEN3C	0.8000	0.5333	0.8830	0.8246	0.4408	0.9882
SeedDream 4.0 + Perception-as-Control	0.6667	0.3167	0.9188	0.8874	0.4799	0.9859
AdDirector (Ours)	0.8833	0.5667	0.9571	0.9349	0.4540	0.9945

Engine 5. TrajectoryCrafter [60] relies on depth estimation, dynamic point-cloud rendering, and user-specified target camera trajectories. In contrast, AdDirector requires only a product reference image and a textual camera instruction, without multi-view synchronized videos, camera extrinsics, depth maps, point-cloud rendering, or explicit target trajectories. Overall, these comparisons demonstrate that AdDirector achieves a better balance between semantic camera-motion adherence and product consistency under a substantially lighter input and deployment setting.

Additional Aesthetic Evaluation. The generic Aes.Qual. metric may not fully reflect the quality requirements of product-advertising videos, which emphasize product preservation and controllable camera motion, including logo consistency, texture fidelity, color stability, and geometric completeness. Moreover, relying on a single aesthetic model may introduce evaluator bias. Therefore, we additionally report a GPT-based aesthetic score that is better aligned with the quality criteria of this specific scenario. As shown in Tab. 7, AdDirector achieves a score of 8.420 and ranks in the middle among the compared methods, with only a small gap from the higher-ranked approaches. This complementary evaluation indicates that the relatively lower generic Aes.Qual. score should not be directly interpreted as a degradation in the visual quality of the generated videos.

D.3 More Qualitative Results

Visual Results across Camera Directions. Fig. 10 presents additional visual results under diverse camera directions, including dolly in/out, pan left/right, and clockwise/counterclockwise orbit.

Qualitative Comparisons with Explicit Camera-control Methods. Fig. 11 presents additional qualitative comparisons with explicit camera-control methods. We observe that explicit camera-control models [3, 60] can provide relatively stable and direct control over camera-motion directions. However, they may also introduce product contour distortion, aspect-ratio deformation, text/logo drift, and texture inconsistency, especially under dolly and orbit motions. In advertisement video generation, however, product fidelity is the primary requirement, and camera motion should serve the final product presentation. Additionally, TrajectoryCrafter [60] uses point-cloud renderings from 3D geometric reprojection as camera conditions, which requires reliable depth/geometry estimation and a predefined target camera trajectory. When these geometric conditions do not fully capture the scene structure, the induced motion may become spatially uneven. The foreground subject often follows the

Table 7: Additional aesthetic evaluation using a GPT-based evaluator that is better aligned with the quality criteria of product-advertising videos. A higher score indicates better aesthetic quality.

Rank	Model	GPT-based Aes.Qual.
1	Pika2.5	8.670
2	Kling1.6	8.645
3	VACE 14B	8.490
4	Phantom	8.470
5	**AdDirector (Ours)**	**8.420**
6	SkyReels-A2	8.380
7	HunyuanCustom	8.155
8	MAGREF	8.115
9	BindWeave	7.840

Table 8: Ablations on the key hyperparameters and design choices of TAR. GT denotes the ground-truth flow-matching loss, and I denotes the interval between supervised DiT blocks.

	Camera Acc.		Subject Cons.		Video Qual.	
Setting	**Cam.IS↑**	**VQAScore↑**	**CLIP-I↑**	**DINO-I↑**	**Aes.Qual.↑**	**Motion↑**
Balancing weight						
TAR 0 / GT 1	0.485	0.3700	0.9248	0.8781	0.4283	0.9924
TAR 0.25 / GT 0.75	0.695	0.4400	0.9326	0.8925	0.4282	0.9929
TAR 0.5 / GT 0.5 (default)	0.740	0.4450	0.9304	0.8872	0.4275	0.9938
TAR 0.75 / GT 0.25	0.635	0.4250	0.9236	0.8730	0.4359	0.9941
Block-layer interval						
$I = 1$ (40 blocks)	OOM	OOM	OOM	OOM	OOM	OOM
$I = 2$ (20 blocks)	0.760	0.5050	0.9278	0.8837	0.4313	0.9936
$I = 4$ (10 blocks)	0.725	0.4700	0.9195	0.8658	0.4333	0.9929
$I = 8$ (5 blocks, default)	0.740	0.4450	0.9304	0.8872	0.4275	0.9938
$I = 10$ (4 blocks)	0.695	0.4550	0.9300	0.8903	0.4343	0.9936
Supervision strategy						
Block features (default)	0.740	0.4450	0.9304	0.8872	0.4275	0.9938
Gradient	0.550	0.3900	0.9269	0.8806	0.4333	0.9926

camera motion strongly, while the background shows limited motion variation, thereby weakening the perceived spatial depth and overall camera dynamics of the generated video.

Additional Comparisons with Reference-to-Video Methods. Figs. 12, 13, 14, and 15 present additional qualitative comparisons against existing methods [14, 16, 23, 27, 29, 33, 37, 41] across a range of camera motion types. The results demonstrate that AdDirector consistently achieves strong adherence to diverse semantic camera instructions while maintaining visual consistency of product appearance throughout the motion. In contrast, existing state-of-the-art approaches exhibit notable limitations in simultaneously preserving both camera controllability and appearance fidelity across varying camera movements, further underscoring the robustness of AdDirector.

D.4 More Ablation Study

Base-model Baseline. The first row of Tab. 2 corresponds to Wan 2.2 fine-tuned on our advertisement dataset without either STS or TAR. This baseline isolates the effect of dataset-specific fine-tuning from the contributions of the proposed modules. Its camera accuracy remains below 50%, indicating that fine-tuning on advertisement data alone is insufficient to reliably follow camera instructions. Adding STS and TAR consistently improves quantitative performance and qualitative results, demonstrating their contributions beyond dataset-specific fine-tuning.

Table 9: Ablations on the key design choices of STS. The default configuration uses three slots, a CLS token, segmented self-attention, and a mixed-scale candidate set.

	Camera Acc.		Subject Cons.		Video Qual.	
Setting	**Cam.IS↑**	**VQAScore↑**	**CLIP-I↑**	**DINO-I↑**	**Aes.Qual.↑**	**Motion↑**
$N = 3$ (first/middle/last, default)	0.695	0.4500	0.9306	0.8964	0.4376	0.9934
$N = 2$ (first/last)	0.630	0.4650	0.9289	0.8782	0.4282	0.9935
No CLS	0.645	0.4050	0.9301	0.8829	0.4449	0.9932
No segmented self-attention	0.665	0.4550	0.9208	0.8686	0.4481	0.9933
Only crop ($S = 5$)	0.645	0.4150	0.9279	0.8790	0.4332	0.9945
Only downsample ($S = 3$)	0.650	0.4300	0.9283	0.8971	0.4405	0.9938

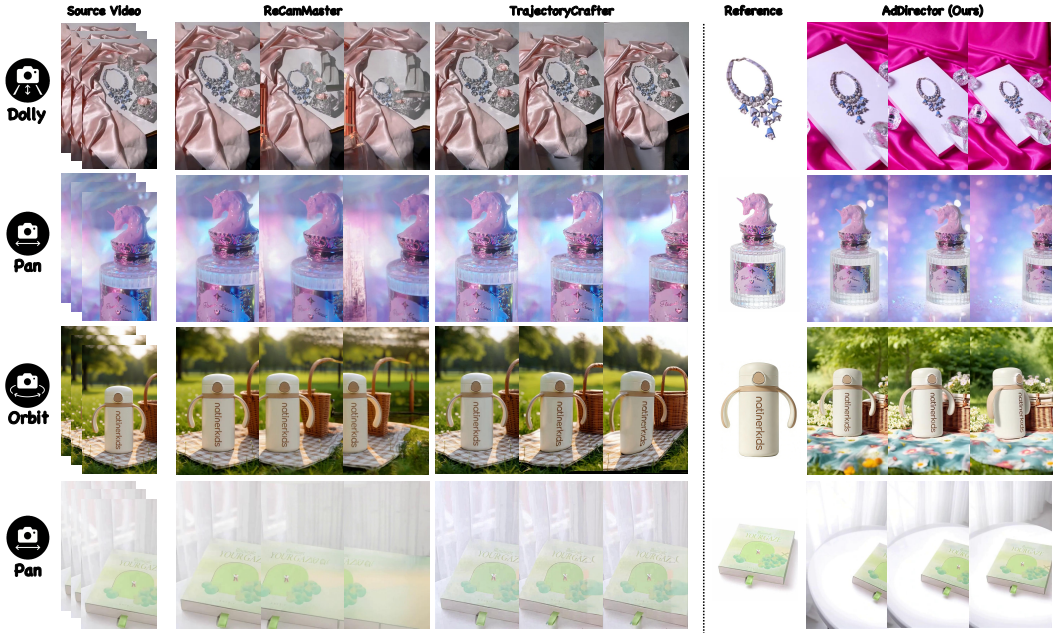

Figure 11: Qualitative comparisons with explicit camera-control methods.

Computational Cost. Tab. 2 further reports computational cost. It shows that TAR introduces no additional inference cost, while STS incurs a marginal increase of approximately 0.8B parameters, accompanied by only slight growth in memory usage and single-GPU inference time. These findings indicate that our framework substantially improves camera accuracy and subject consistency in R2V generation without introducing significant computational overhead.

Additional Ablations on Key Hyperparameters. We further conduct comprehensive ablations on the key hyperparameters and design choices of the TAR and STS modules.

- **1. TAR module.** We investigate the balancing weight, block-layer interval, and supervision strategy of TAR, with the results summarized in Tab. 8. **(1) Balancing weight.** TAR denotes the weight of the TAR loss, while GT denotes the weight of the ground-truth flow-matching loss. The two losses are complementary: using the GT loss alone is insufficient, whereas an overly large TAR weight imposes excessive temporal regularization that harms camera adherence and subject consistency. Therefore, TAR 0.5 / GT 0.5 provides a reasonable default balance. **(2) Block-layer interval.** The interval I controls the effective regularization density of temporal anchoring rather than merely the number of supervised blocks. Denser supervision can improve camera accuracy, but increasing the number of supervised blocks is not monotonically beneficial. Overly dense supervision increases memory overhead, while some intermediate densities introduce additional optimization interference. Therefore, $I = 8$ provides a practical trade-off among efficiency, stability, and overall quality. **(3) Supervision**



Figure 12: More qualitative results of existing methods for **static** camera motion.

strategy. We compare block-feature alignment with gradient supervision. Directly aligning the intermediate block features of the anchor and main branches is more effective than gradient supervision.

- **2. STS module.** We investigate the number of STS slots, the CLS token, segmented self-attention, and the construction of the multi-scale candidate set, with the results summarized in Tab. 9. **(1) Slot number.** $N = 3$ attains the highest Cam.IS while maintaining strong CLIP-I and DINO-I, showing that a moderate increase in the number of slots captures richer reference features and improves both camera-instruction adherence and product-structure consistency. $N = 2$ is less stable than $N = 3$, while a larger N increases the attention cost. Therefore, $N = 3$ provides a sound performance–cost trade-off. **(2) CLS token.** Removing the CLS token markedly lowers VQAScore, indicating that it facilitates global reference-information aggregation. **(3) Segmented self-attention.** Removing segmented self-attention degrades CLIP-I and DINO-I, demonstrating its importance in facilitating interaction between each temporal segment and its corresponding STS slot. **(4) Scale candidates.** Using only cropped or only downsampled candidates lowers Cam.IS, confirming that either source alone is insufficient to stably support camera-aware reference injection. In contrast, the mixed candidate set combines local details with global structure.

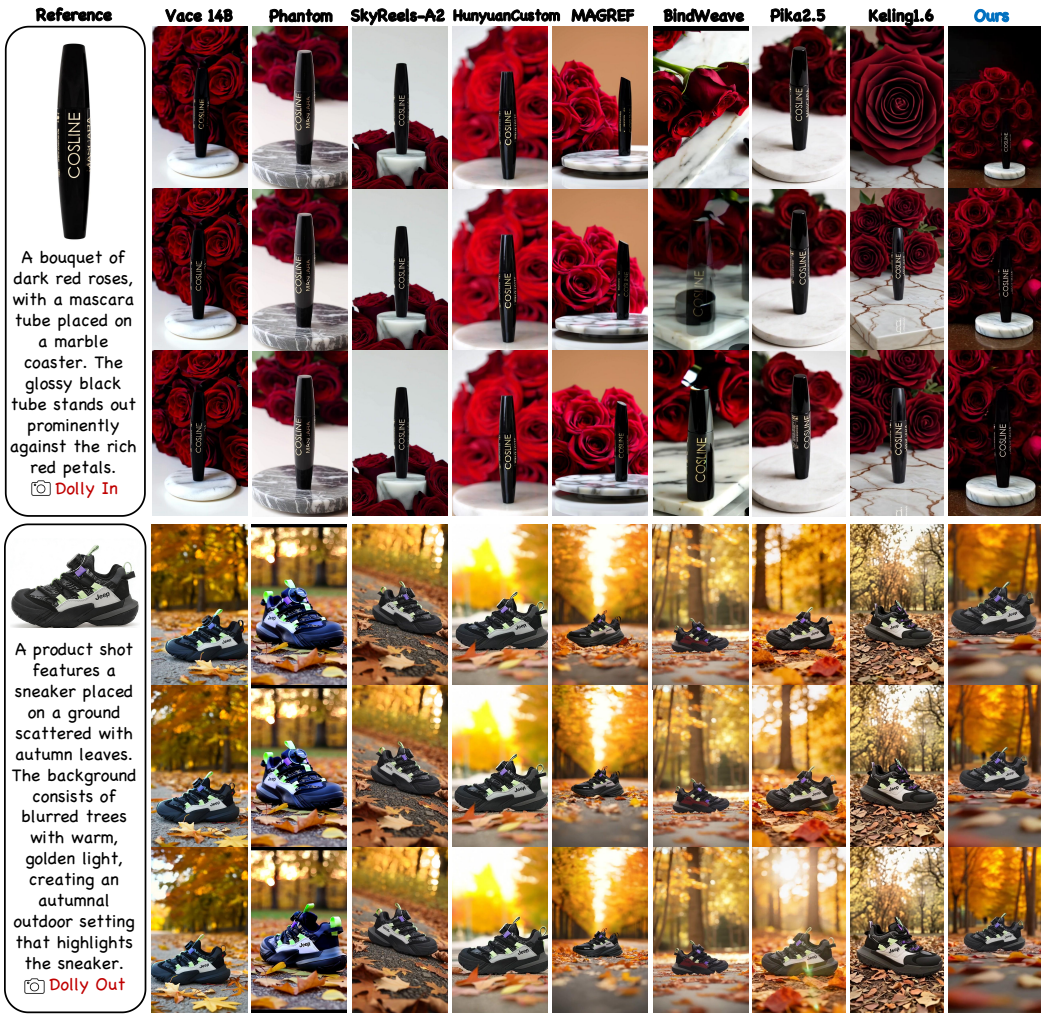

Figure 13: More qualitative results of existing methods for **dolly** camera motion.

D.5 User Study

We conducted a user study with 30 participants to evaluate the results. For each participant, 20 videos were randomly sampled, covering all camera motion types proportionally. For each video, participants selected the best-generated result along three dimensions: camera accuracy, video quality, and subject consistency. The aggregated preference ratio for each dimension was then computed and reported as the human evaluation score. As shown in Tab. 1, the results indicate that our method attains the highest preference ratios across all three dimensions. To ensure consistent evaluation criteria across participants, we provide the assessment guidelines summarized in Tab. 4, which specify the aspects participants should attend to when selecting the best video for each evaluation dimension.

E Limitations

While AdDirector demonstrates substantial improvements in generating camera-controllable advertisement videos with high product fidelity, it still presents certain limitations. First, although AdDirector excels at translating high-level semantic camera instructions into coherent spatio-temporal dynamics, achieving precise control over highly customized, multi-stage, and non-standard geometric trajectories relying solely on textual prompts remains challenging. Furthermore, current state-of-the-art automated evaluation metrics still frequently misclassify correct camera motions. This highlights the critical need for developing more robust, camera-aware video quality assessment benchmarks.

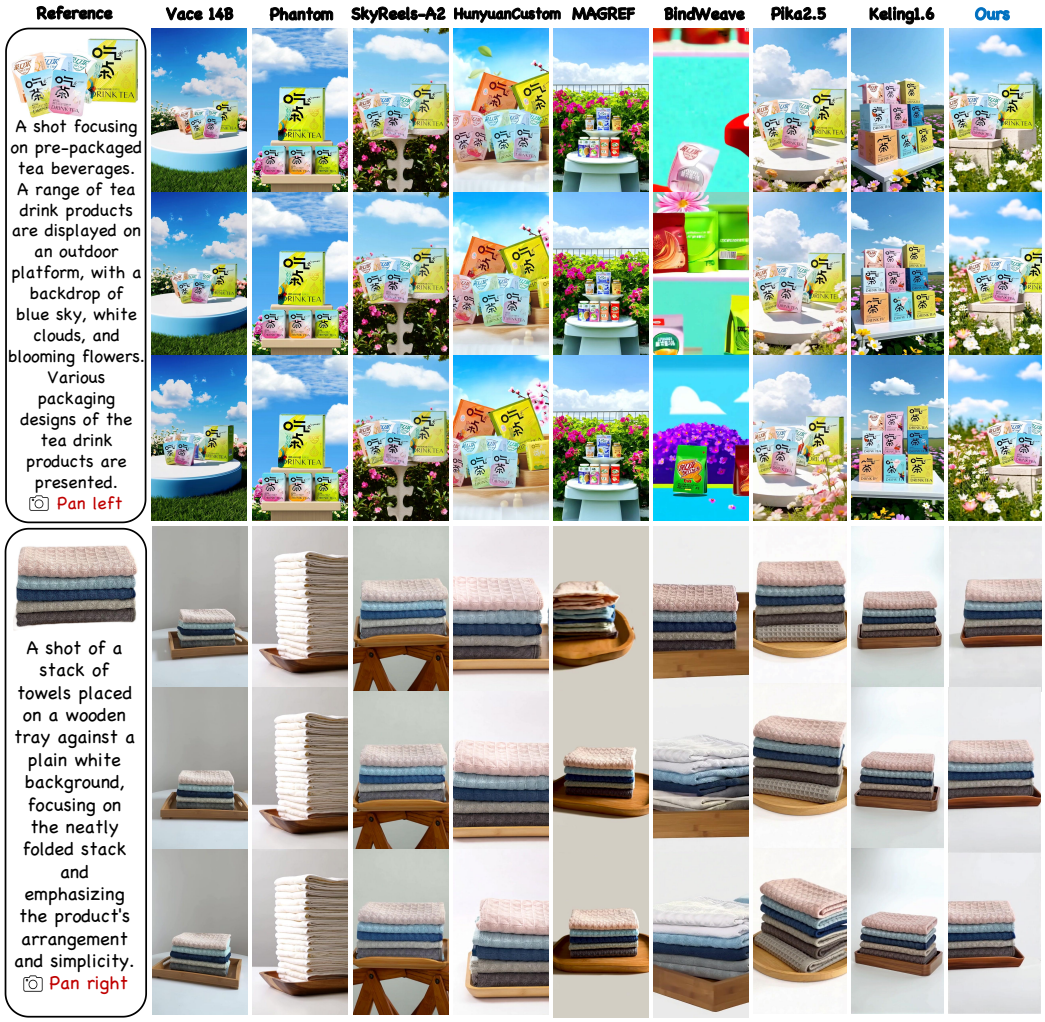

Figure 14: More qualitative results of existing methods for **pan** camera motion.

F Social Impact

Our proposed framework, AdDirector, significantly advances reference-to-video (R2V) generation with precise camera control and high subject fidelity. While this can democratize video production in the advertising and creative industries by reducing reliance on expensive equipment and lowering the barrier to entry for creators, it also presents potential societal risks. Like other advanced generative models, AdDirector could be misused to create misleading content, deepfakes, or unauthorized advertisements. To mitigate these risks, it is crucial to establish strict usage guidelines, implement robust watermarking or deepfake detection, and prioritize training data transparency to protect intellectual property rights.

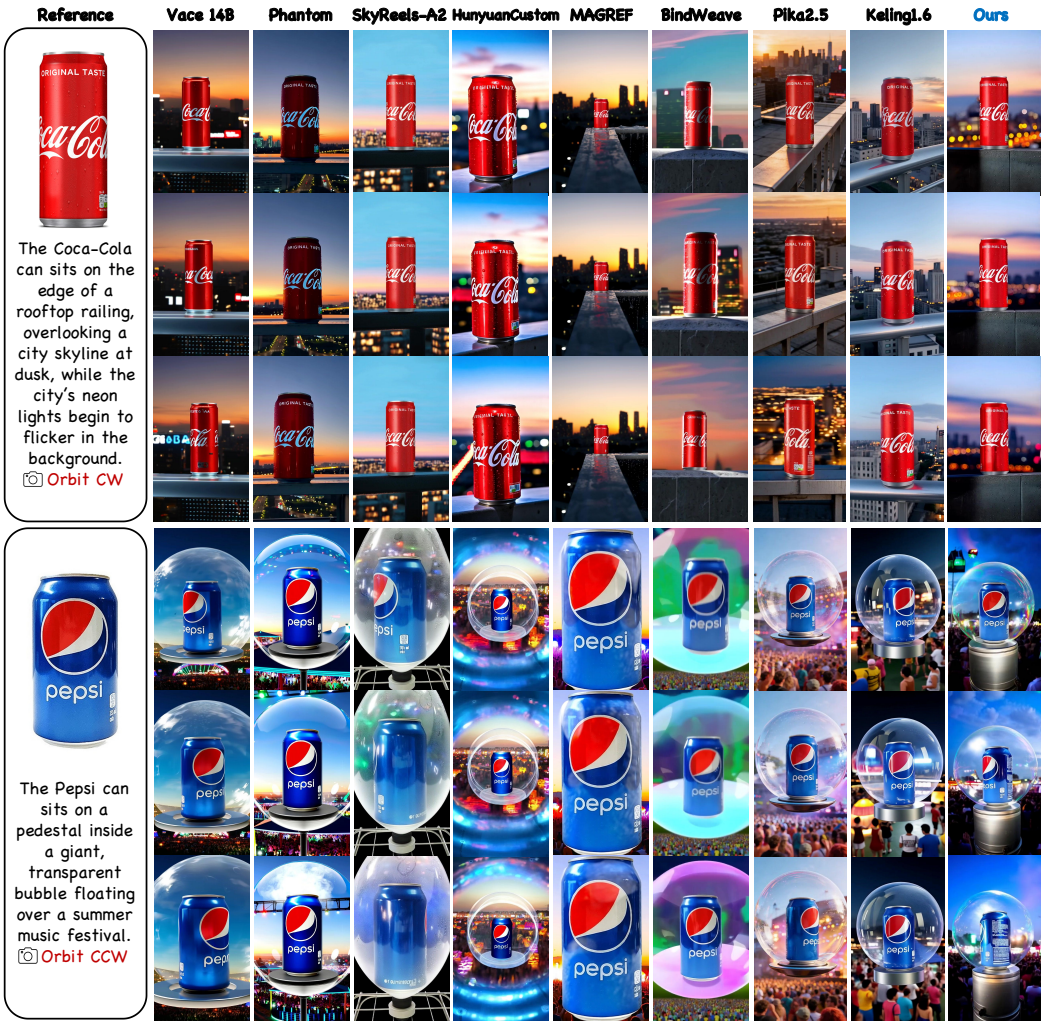

Figure 15: More qualitative results of existing methods for **orbit/rotate** camera motion.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately summarize the proposed method (AdDirector) and our experimental findings.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We explicitly discuss the limitations of our proposed method and potential directions for future research in Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper focuses on a deep learning framework for video generation and does not include theoretical derivations or proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide full training details, architecture, and evaluation protocol.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We will release code upon acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: The experimental settings, including implementation details, hyperparameters, and dataset curation, are clearly specified in Section 4.1 and Appendix D.1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [N/A]

Justification: The paper evaluates video generation models using automated metrics (e.g., VQAScore, FVD) and human evaluations, which typically do not rely on error bars or statistical significance tests.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper explicitly states the compute resources used (e.g., experiments and fine-tuning were conducted on 32 NVIDIA H20 GPUs with a batch size of 1).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss both positive applications (content creation) and risks (deepfake misuse) in Appendix F.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not publicly release the generated models or the newly collected dataset, so no specific safeguards for public release are required.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper properly cites and credits all existing methods, models, and tools used in the research (e.g., TransNet v2, Qwen-VL, GroundingDINO, SAM2, and CameraBench).

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: A user study with 30 participants was conducted to evaluate the generated videos. Full details and evaluation guidelines given to participants are provided in Appendix D.5 and Table 4.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The human subjects research was strictly limited to a perceptual visual quality evaluation (user study) of generated videos, which does not require IRB approval.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The paper explicitly declares the use of Qwen-VL to automatically generate textual descriptions for the dataset curation process in Appendix B.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.