InstructVVT: Instruction-Driven Video Virtual Try-On without Auxiliary Spatial Priors

Dingbao Shao[1,2,*], **Song Wu**[2,*,‡], **Xinyu Chen**[1,2], **Qian Wang**[2], **Jiahang Li**[1], **Kuai Jiang**[1]
Jiang Lin[1], **Yuhang Liu**[1], **Ziyu Chen**[1], **Duo Li**[1], **Jiaxin Hu**[1], **Shengrong Gu**[1]
Ziheng Tang[1], **Rongrong Liu**[1], **Yanlun Peng**[3], **Liang Li**[2], **Junlan Feng**[2]
Lujia Jin[2], **Ting Zhang**[2], **Jian Yang**[1], **Zili Yi**[1,†]
[1]Nanjing University, China
[2]China Mobile Jiutian Artificial Intelligence Technology (Beijing) Co., Ltd.
[3]Independent Researcher
[*]Equal contribution. [†]Corresponding author. [‡]Project leader.

Abstract

Video virtual try-on is a highly constrained editing task requiring the precise replacement of a target person's clothing while strictly preserving the original video's spatial structure and temporal dynamics. Existing methods heavily rely on auxiliary handcrafted spatial priors (e.g., masks, poses) for editing control. However, these priors are prone to failure in unconstrained real-world videos and often compress rich visual context into incomplete structural signals. Furthermore, standard reconstruction objectives fail to fully capture try-on-specific human preferences. To address these challenges, we propose InstructVVT, an instruction-driven and reference-guided video virtual try-on framework based on a Diffusion Transformer (DiT) that operates without inference-time spatial priors. Our core insight is to recover fine-grained control directly from the input triplet (source video, reference garment, and instruction) via a dual-level reference conditioning scheme. Specifically, an MLLM infers semantic edit tokens for target disambiguation and structural preservation, while a lightweight conditioning pathway explicitly injects fine-grained visual garment details. Finally, we design a try-on-specific reward and utilize the DiffusionNFT algorithm to align the model with human preferences. Extensive experiments on ViViD-S and TripVVT-Bench demonstrate that InstructVVT outperforms state-of-the-art open-source methods in garment fidelity, structural preservation, and temporal consistency, despite requiring fewer inference-time controls.

1 Introduction

Video virtual try-on constitutes a video editing task with reference constraints and structural preservation properties [Fang et al., 2024, Zuo et al., 2025, Li et al., 2025a, Shao et al., 2026]. Taking a source video and a reference garment image as inputs, the task requires the model to replace the clothing of the target person, while maintaining the consistency of personal identity, body shape, pose dynamics, camera movement, scene layout, background content and temporal coherence. Different from general video generation tasks, visual realism is far from sufficient for video virtual try-on. Specifically, the edited video should faithfully restore the style and texture of the reference garment, accurately apply the clothing replacement to the designated human subject, and keep irrelevant scene content completely unchanged.

The core difficulty of video virtual try-on is to provide the model with precise editing control: it must know who to edit, where to edit, how to transfer the garment, and what to preserve. Existing methods typically obtain such control from auxiliary spatial priors [Zhu et al., 2023, Kim et al.,

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

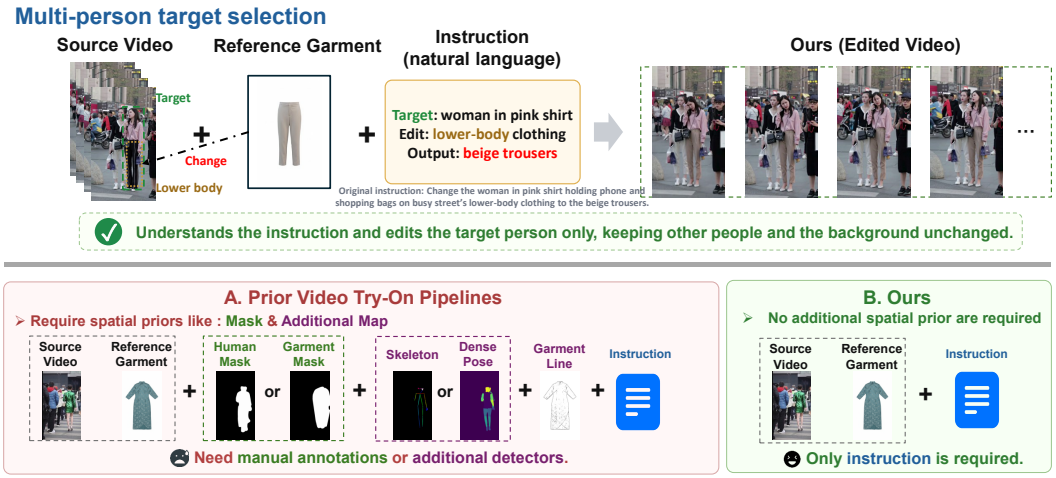


Figure 1: Teaser examples of instruction-driven video virtual try-on. Given a source video, a reference garment, and a natural-language instruction, the model edits the instructed target person while preserving the source identity, motion, background, and temporal consistency.

2023, Choi et al., 2024, Fang et al., 2024, Zuo et al., 2025, Li et al., 2025a, Shao et al., 2026], including human masks, garment masks, parsing maps, DensePose representations, skeletons, or garment structure lines. These priors provide explicit localization and preservation cues, enabling high-quality try-on in constrained settings. However, three limitations stand out. (i) Their performance depends heavily on external estimators or manual preprocessing, which can be unreliable in unconstrained videos with unusual viewpoints, occlusions, loose garments, large camera motion, or multiple people. (ii) Since these controls are predefined and task-specific, they may compress the source video and reference garment into incomplete structural signals, losing useful visual context for target disambiguation, garment-to-body alignment, and non-edited region preservation. (iii) Supervised reconstruction or denoising objectives alone cannot fully capture the qualities that define a good try-on result, and may lead to weakened garment details, unintended changes to non-target regions, incorrect target editing, or temporal flicker. Recently, general instruction-guided video editing models have shown strong editing capabilities [Molad et al., 2023, Ku et al., 2024b, Wei et al., 2025]. Nevertheless, they are not specifically optimized for the fine-grained constraints of video virtual try-on, and often struggle with reference-garment fidelity, target correctness, background preservation, and temporal stability.

To address these challenges, we propose InstructVVT, a DiT-based video diffusion framework for instruction-driven and reference-guided video virtual try-on, as shown in Fig. 2. InstructVVT takes a source video, a reference garment image, and a natural-language instruction as inputs, and generates an edited video in which the instructed person wears the reference garment. During inference, it does not require auxiliary handcrafted spatial priors. Instead, InstructVVT recovers editing control from the input triplet itself. Specifically, we use a multimodal large language model (MLLM) to jointly parse the source video, the reference garment, and the instruction, and extract garment-aware edit-intent tokens to guide the video diffusion model. Compared with handcrafted spatial priors, these edit-intent tokens provide a more flexible semantic interface: they help resolve the target subject, the editable clothing region, the preservation constraints, and the high-level alignment between the garment and the person. In parallel, source-video latents are fed into the generator to anchor motion, layout, camera behavior, and non-edited regions. To improve reference-garment fidelity, we further design a lightweight garment conditioning branch that injects fine-grained visual garment tokens into the Diffusion Transformer. This dual use of the reference garment is important: the MLLM provides semantic garment-to-person alignment, while the generator-side garment tokens preserve texture, pattern, shape, and visible design details. Together, these conditions allow the generator to treat video virtual try-on as a localized edit of the source video, rather than synthesis driven by externally specified spatial maps.

In addition, inspired by recent MLLM-based feedback methods [Li et al., 2025b], we design a training-free video try-on reward to address the limitation that supervised losses do not fully char-

acterize try-on quality. The reward evaluates generated videos from try-on-specific aspects, including garment fidelity, instruction following, target correctness, identity and background preservation, motion preservation, and temporal consistency. We combine this reward with DiffusionNFT algorithm [Zheng et al., 2025] to align the generator with human try-on preferences.

Experiments on ViViD-S [Chong et al., 2025] and TripVVT-Bench [Shao et al., 2026] show that InstructVVT outperforms current open-source state-of-the-art video try-on models across evaluation metrics and human preference, while requiring no auxiliary spatial controls at inference time. Our contributions are threefold:

- We propose InstructVVT, a DiT-based video diffusion framework that establishes an instruction-driven and reference-guided video virtual try-on paradigm without relying on inference-time auxiliary handcrafted spatial priors.
- We present a dual-level reference conditioning scheme for DiT-based video try-on: an MLLM jointly analyzes the source video, reference garment, and instruction for garment-aware edit planning, while a lightweight garment-token pathway provides fine-grained visual appearance details.
- We design a try-on-specific reward based on a frozen MLLM and use DiffusionNFT-based post-training to improve garment fidelity, target correctness, source preservation, motion naturalness, and temporal consistency.

2 Related Work

2.1 Virtual try-on with spatial priors

Virtual try-on aims to transfer a reference garment to a target person while preserving identity, pose, body shape, and background content. Image virtual try-on has been studied extensively with warping-based, parsing-based, and diffusion-based frameworks [Zhu et al., 2023, Kim et al., 2023, Choi et al., 2024, Chong et al., 2024], while video virtual try-on further requires temporal consistency under body motion, camera movement, and occlusion [Fang et al., 2024, Chong et al., 2025, Zuo et al., 2025, Li et al., 2025a, Shao et al., 2026]. To achieve controllable editing, many existing methods rely on explicit spatial priors, such as human masks, garment masks, parsing maps, poses, DensePose-like representations, skeletons, or garment structure lines. These signals provide useful localization and preservation cues, but also introduce dependence on external estimators or manual preprocessing. In challenging in-the-wild videos, such priors may be inaccurate, temporally unstable, or unavailable.

Recent mask-free or reduced-condition try-on methods improve usability by removing some spatial inputs, especially garment or edit-region masks [Chang et al., 2024, Zhang et al., 2025, Shen et al., 2025, Wan et al., 2025]. However, they often still depend on other structural conditions or assume a relatively fixed try-on interface. This makes it difficult to handle natural-language instructions that specify the target person, edit scope, and preservation requirements jointly. In contrast, our method removes inference-time auxiliary structural priors and uses only a source video, a reference garment, and an instruction.

2.2 Instruction-guided and multimodal visual editing

Instruction-guided image and video editing methods provide a more flexible interface than task-specific controls [Brooks et al., 2023, Zhang et al., 2023, Molad et al., 2023, Ku et al., 2024b]. By conditioning generation on natural-language commands, these models can perform diverse edits without manually specifying low-level spatial annotations. More recently, MLLM-guided editing systems use multimodal representations to interpret visual context together with the user instruction [Fu et al., 2024, Pan et al., 2025, Mou et al., 2025, Wei et al., 2025], which is especially useful when the target object is described by attributes, relations, or scene context rather than by an explicit mask.

However, video virtual try-on has stricter constraints than generic visual editing. A successful try-on result must preserve the source identity, body motion, scene layout, and non-target regions, while faithfully transferring the reference garment and maintaining temporal stability. General editing models may follow the instructions but drift from the garment reference, edit the wrong person, alter

the background, or introduce inconsistent clothing details across frames. Our work adapts MLLM-based conditioning to video try-on by using the MLLM to jointly parse the source video, reference garment, and instruction into edit-intent tokens, while also injecting the garment appearance through a generator-side visual branch.

2.3 Reward alignment for diffusion models

Most diffusion-based editing and try-on models are trained with reconstruction, denoising, or flow-matching objectives. These objectives are effective for learning paired data distributions, but they do not directly optimize human preferences. In virtual try-on, important criteria such as garment fidelity, target correctness, source preservation, visual naturalness, and temporal consistency are difficult to capture with a single supervised loss.

Reward-based post-training provides a way to align diffusion models with such high-level preferences [Fan et al., 2023, Black et al., 2024, Wallace et al., 2024, Zheng et al., 2025]. Instead of relying only on pixel-level or feature-level reconstruction, a reward model can evaluate generated samples according to task-specific criteria and guide the generator toward preferred outputs. Recent instruction-based editing work also explores training-free MLLM implicit feedback as a reward signal [Li et al., 2025b]. In this work, we construct a training-free MLLM score-token reward for video try-on and use DiffusionNFT-based post-training to improve try-on-specific qualities after supervised fine-tuning.

3 Method

3.1 Overview

As shown in Fig. 2, our framework is built on an open source latent video Diffusion Transformer (DiT) [Team Wan et al., 2025, Peebles and Xie, 2023] and an MLLM. Given a source video V_{s}, a reference garment image I_{g}, and a natural-language instruction $\mathcal{I}$, our model generates an edited video $\hat{V}$ in which the instructed person wears the reference garment while the remaining content is preserved. At inference time, our model uses only these three inputs and does not require auxiliary structural annotations such as masks, poses, parsing maps, DensePose-like conditions [Güler et al., 2018], skeletons, or garment contours.

Our model comprises three complementary conditioning pathways. First, latents encoded from the source video by a pretrained VAE preserve motion, layout, and non-edited regions. Second, a frozen MLLM with trainable LoRA parameters [Hu et al., 2022] takes sampled frames from the source video, reference garment and instruction as inputs. It generates garment-aware edit tokens to locate the target subject, define the edit scope, and establish high-level alignment between the garment and the person. Third, reference-garment tokens provide explicit visual appearance via a generator-side pathway. The latter two garment-aware pathways are complementary: the MLLM branch enhances semantic alignment with the reference garment, while the generator-side garment tokens retain fine-grained texture, pattern, shape and subtle design details.

We adopt two major training paradigms: supervised fine-tuning and reinforcement learning training. The supervised fine-tuning paradigm learns the correspondence among source video, reference garment, instruction, and target try-on video. The reinforcement learning stage employs a frozen MLLM [Bai et al., 2025b] as the reward model and leverages the DiffusionNFT [Zheng et al., 2025] algorithm to align the generator with specific preferences for virtual try-on. Experimental details, stage-wise training settings, and compute resources are provided in Appendix C.

3.2 Instruction-Driven Video Virtual Try-On (InstructVVT)

MLLM Edit Tokens. A text-only instruction is often insufficient for video try-on. As shown in Fig. 1, instructions like "change the lower-body clothing of the woman in the pink shirt to beige trousers." require the model to understand the source video. It needs to detect all people, infer their spatial relations, locate clothing regions, and identify content that should stay unchanged. Moreover, a single text instruction cannot fully describe the fine-grained garment appearance, nor clarify how the reference garment aligns with the target body. We thus leverage an MLLM to jointly process sampled source frames, the reference garment image, and the instruction $\mathcal{I}$. The MLLM infers the

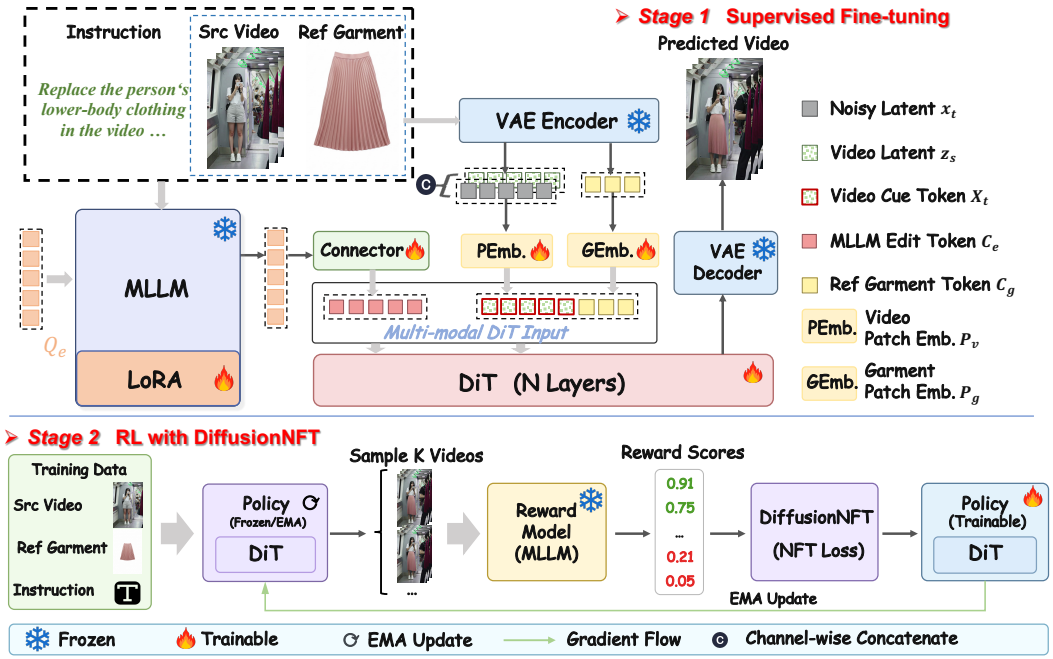


Figure 2: Overview of our InstructVVT framework. In the supervised training, the source video, reference garment, and instruction are encoded by a frozen MLLM with trainable query tokens Q_e, LoRA adapters [Hu et al., 2022], and an MLP connector to produce MLLM edit tokens C_e. We adopt two embedding layers with identical architecture but independent parameters to encode video cue tokens X_t and reference garment tokens C_g, respectively. Video cue tokens X_t and reference garment tokens C_g are concatenated along the token dimension and jointly participate in DiT self-attention, while MLLM edit tokens C_e provide cross-attention conditioning in each DiT block. The post-training stage applies DiffusionNFT [Zheng et al., 2025]: a frozen EMA policy samples K videos, a tailored reward model based on MLLM assigns score-token rewards, and the NFT loss updates the trainable DiT.

editing context, including the target person, editing scope, garment-body alignment, and content preservation constraints.

Following recent methods [Pan et al., 2025, Mou et al., 2025], we introduce learnable query tokens Q_{e} to extract compact editing features from the MLLM. The query tokens are appended to the multimodal prompt and are processed together with the source frames, reference garment, and instruction. We then select the hidden states at the query-token positions and project them to the DiT hidden dimension:

$$C_{\mathrm{e}} = P_{\mathrm{e}}\left(\mathrm{Select}_{Q}\left[\mathrm{MLLM}(V_{\mathrm{s}}, I_{\mathrm{g}}, \mathcal{I}, Q_{\mathrm{e}})\right]\right), \tag{1}$$

where P_{e} is an MLP connector and C_{e} denotes the MLLM edit tokens.

Since the MLLM observes the reference garment, the edit tokens encode not only source-aware target localization and edit scope, but also high-level garment semantics useful for aligning the reference clothing to the target person. These tokens provide semantic and relational guidance rather than explicit pixel-level masks, poses, parsing maps, or edit regions.

Video Cue Tokens and Garment Tokens. Let x_t be the noisy video latent at diffusion timestep t, and let $z_{\mathrm{s}} = \mathcal{E}_{\mathrm{vae}}(V_{\mathrm{s}})$ be the source-video latent. The source latent z_t is concatenated with the noisy latent x_t along the channel dimension and fed into the video patch embedding layer to yield video cue tokens X_t:

$$X_t = P_{\mathrm{v}}\left(\mathrm{Concat}_{\mathrm{ch}}(x_t, z_{\mathrm{s}})\right), \tag{2}$$

where P_{v} is the video patch embedding layer. Since the pretrained patch embedding expects only target-latent channels, we expand its input channel dimension by a factor of two. The original

channel weights are initialized from the pretrained model, and the newly added source-channel weights are initialized to zero for stable fine-tuning.

The reference garment is encoded by the VAE and mapped into garment tokens C_{g} via an independent garment patch embedding layer:

$$C_{\mathrm{g}} = \mathrm{Flatten}\left(P_{\mathrm{g}}\left(\mathcal{E}_{\mathrm{vae}}(I_{\mathrm{g}})\right)\right), \tag{3}$$

where P_{g} denotes the garment patch embedding layer and C_{g} denotes the reference-garment tokens. This generator-side garment pathway provides fine-grained visual evidence for synthesis and does not use garment masks, contours, parsing maps, or structure-line extractors.

Condition Injection within DiT. The video tokens X_t and garment tokens C_{g} are concatenated along the token dimension before entering the DiT:

$$H_0 = \mathrm{Concat}_L\left(X_t, C_{\mathrm{g}}\right). \tag{4}$$

We replace the original T5 text embeddings of the base DiT [Team Wan et al., 2025] with our MLLM edit tokens C_{e}. These tokens are injected into each DiT block via cross-attention:

$$H_{\ell+1} = \mathrm{DiTBlock}_{\ell}\left(H_{\ell}, t, C_{\mathrm{e}}\right), \qquad \ell = 0, \ldots, L_{\mathrm{dit}} - 1. \tag{5}$$

After the final DiT block, only the video-token positions are unpatchified to predict the target latent; the garment tokens serve as conditioning tokens and are discarded before VAE decoding.

This design lets the model recover try-on controllability from the input triplet itself. The MLLM edit tokens provide instruction- and reference-aware editing context, the source latent provides dense structural anchoring, and the garment tokens provide explicit appearance cues, all without inference-time auxiliary spatial priors.

3.3 Two major training paradigms

Supervised Fine-Tuning. Given a paired example $(V_{\mathrm{s}}, I_{\mathrm{g}}, \mathcal{I}, V_{\mathrm{t}})$, we encode the target video as $x_0 = \mathcal{E}_{\mathrm{vae}}(V_{\mathrm{t}})$ and optimize the standard flow-matching objective [Lipman et al., 2023]:

$$\mathcal{L}_{\mathrm{SFT}} = \mathbb{E}_t\left[\left\| f_\theta\left(x_t, z_{\mathrm{s}}, C_{\mathrm{g}}, C_{\mathrm{e}}, t\right) - v \right\|_2^2\right], \tag{6}$$

where x_t and v are produced by the flow-matching noise schedule. This stage teaches the model the basic correspondence among the source video, reference garment, instruction, and target try-on video. In this stage, the MLLM backbone is kept frozen, while the query-token embeddings, LoRA adapters, and connector are trainable.

Reinforcement Learning with Try-On-Specific Reward. Supervised fine-tuning does not directly optimize the preference criteria that matter for try-on, such as reference-garment fidelity, correct target editing, preservation of non-edited regions, natural motion, and temporal stability. We therefore use a frozen pretrained MLLM as a training-free reward model. This reward MLLM is separate from the LoRA-adapted MLLM used for generator conditioning.

Given $(V_{\mathrm{s}}, I_{\mathrm{g}}, \mathcal{I}, \hat{V})$, the reward MLLM is prompted with a fixed try-on rubric and asked to directly output one score from 0 to 5. Instead of parsing generated text, we compute a score-token reward from the first-token logits. Let ℓ_j be the logit of score token $j \in \{0, 1, 2, 3, 4, 5\}$. We define

$$p_j = \frac{\exp(\ell_j)}{\sum_{m=0}^{5} \exp(\ell_m)} \qquad \text{and} \qquad \rho(\hat{V}, c) = \frac{1}{5}\sum_{j=0}^{5} j\, p_j, \tag{7}$$

where $c = (V_{\mathrm{s}}, I_{\mathrm{g}}, \mathcal{I})$ and $\rho \in [0, 1]$. This non-CoT score-logit reward avoids brittle text parsing and provides a smooth scalar signal for post-training.

For each condition c, a frozen EMA policy samples candidate videos, which are scored by the reward model. We then apply DiffusionNFT [Zheng et al., 2025] to update the trainable generator. Let $r = \rho(\hat{V}, c)$ be the reward of a sampled video. For its latent x_0, we draw a timestep t and compute the noised latent x_t and velocity target v using the same flow-matching schedule. Let $v_{\theta_{\mathrm{old}}}$ and v_θ denote the EMA and trainable velocity predictors. DiffusionNFT constructs

$$v_\theta^{+}(x_t, c, t) = (1 - \beta) v_{\theta_{\mathrm{old}}}(x_t, c, t) + \beta v_\theta(x_t, c, t) \tag{8}$$

and

$$v_\theta^-(x_t, c, t) = (1+\beta)v_{\theta_{\text{old}}}(x_t, c, t) - \beta v_\theta(x_t, c, t). \tag{9}$$

The post-training loss is

$$\mathcal{L}_{\text{NFT}} = \mathbb{E}\left[r \left\|v_\theta^+(x_t, c, t) - v\right\|_2^2 + (1-r)\left\|v_\theta^-(x_t, c, t) - v\right\|_2^2\right]. \tag{10}$$

High-reward samples guide the generator toward preferred try-on behavior, while low-reward samples define directions to avoid. The reward model is used only during post-training and is not required at inference time.

4 Experiments

4.1 Experimental Setup

Training data and protocol. We train InstructVVT on source–reference–target try-on examples from both in-the-wild and indoor domains. The corpus consists of TripVVT-10K [Shao et al., 2026], 10K supplementary ViViD [Fang et al., 2024] video triplets, and around 100K image triplets constructed from public try-on datasets including VITON-HD [Choi et al., 2021] and DressCode [Morelli et al., 2022]; all evaluation videos and garments are excluded from training. Training follows supervised try-on learning followed by DiffusionNFT post-training with the proposed MLLM reward. Appendix C.1 details data construction, Table 6 reports stage-wise training and compute settings, and Appendix C.3 gives the reward-model details and RL reward curve in Fig. 5.

Evaluation benchmarks and metrics. We evaluate on ViViD-S test, following CatV2TON [Chong et al., 2025], and TripVVT-Bench [Shao et al., 2026], which contains more challenging in-the-wild videos with complex motion, diverse scenes, and multi-person cases. Following the TripVVT-Bench protocol, we report metrics covering video quality (VFID$_I$, VFID$_R$), reference-garment fidelity (CLIP-I), background consistency (BG-L1, BG-DINO), temporal consistency (CLIP-F), and reconstruction fidelity when target videos are available (SSIM, LPIPS) [Heusel et al., 2017, Unterthiner et al., 2018, Carreira and Zisserman, 2017, Hara et al., 2018, Radford et al., 2021, Caron et al., 2021, Wang et al., 2004, Zhang et al., 2018]. We additionally report MLLM-Reward because Sec. 4.3 shows its positive correlation with human preference, and its open-source backbone makes the evaluation easier to reproduce. Table 5 provides benchmark sizes and inference settings, while Appendix C.2 describes the reproducibility artifacts.

Baselines. We compare with open-source video virtual try-on models, including ViViD [Fang et al., 2024], CatV2TON [Chong et al., 2025], MagicTryOn [Li et al., 2025a], and TripVVT [Shao et al., 2026], as well as the open-source general video editing model UniVideo [Wei et al., 2025]. On TripVVT-Bench, we additionally include commercial video editing models, including Kling 1.6 [Kuaishou Technology, 2025] and Kling O3 [Kuaishou Technology, 2026], to evaluate against strong general-purpose editors. For methods that require auxiliary spatial priors, we provide the required conditions through their original preprocessing pipelines; Kling 1.6 is evaluated with its available point/region-based editing interface, while Kling O3 uses its video-and-instruction editing interface without region-level control. Additional baseline protocol details are given in Appendix C.2.

4.2 Quantitative Comparisons

Results on ViViD-S test. Table 1 reports quantitative results on ViViD-S test. InstructVVT achieves the best VFID$_I$, VFID$_R$, background metrics, and MLLM-Reward, while maintaining near-best temporal consistency. The discrepancy between generic metrics and MLLM-Reward further shows that video smoothness or generic editing ability alone does not ensure reference-faithful, source-preserving video try-on.

Results on TripVVT-Bench. Table 2 shows results on the more challenging TripVVT-Bench, which contains complex motion, camera changes, and possible target ambiguity. InstructVVT improves the main try-on, reconstruction, garment-fidelity, and human-preference metrics. General editors show complementary limitations: UniVideo is smooth but weak in source preservation, Kling

Table 1: Quantitative comparison with other methods on ViViD-S [Chong et al., 2025] test dataset. We additionally report the proposed MLLM-Reward score. Best results are shown in **bold**, and second-best results are underlined.

Method	VFID$_I$ ↓	VFID$_R$ ↓	CLIP-F ↑	BG-L1 ↓	BG-DINO ↓	MLLM-Reward ↑
ViViD [Fang et al., 2024]	21.8032	0.8212	0.9574	0.0571	0.0055	0.6341
CatV2TON [Chong et al., 2025]	19.5131	0.5283	0.9304	0.0507	0.0071	0.5773
MagicTryOn [Li et al., 2025a]	17.5710	0.5073	0.9647	0.0493	0.0056	0.5199
TripVVT [Shao et al., 2026]	16.4398	0.3315	0.9670	0.0624	0.0059	0.6034
UniVideo [Wei et al., 2025]	23.6970	1.2139	**0.9965**	0.1447	0.0124	0.7152
Ours	**16.0271**	**0.2809**	0.9957	**0.0267**	**0.0048**	**0.7236**

Table 2: Quantitative comparison with other methods on TripVVT-Bench. User preference reports first-place vote ratios from the four-method shortlist study; methods outside the shortlist are marked by "–". Best results are shown in **bold**, and second-best results are underlined; ties share the same rank, with the next distinct value marked as second-best when applicable.

Method	VFID$_I$ ↓	VFID$_R$ ↓	SSIM ↑	LPIPS ↓	CLIP-I ↑	CLIP-F ↑	BG-L1 ↓	BG-DINO ↓	MLLM-Reward ↑	User Pref. ↑
ViViD [Fang et al., 2024]	26.7620	0.7083	0.8323	0.1343	0.8716	0.9849	0.0691	0.0053	0.5569	–
CatV2TON [Chong et al., 2025]	34.2275	3.3266	0.7845	0.2130	0.8077	0.9819	0.1032	0.0109	0.2811	–
MagicTryOn [Li et al., 2025a]	22.9238	0.5502	0.8641	0.1150	0.9102	0.9881	0.0515	**0.0036**	0.5979	5.4%
TripVVT [Shao et al., 2026]	20.7245	0.3163	0.8538	0.1053	0.9373	0.9876	0.0852	0.0059	0.6330	–
UniVideo [Wei et al., 2025]	28.4413	3.0811	0.5461	0.1871	0.9172	**0.9897**	0.2659	0.0091	0.6915	1.6%
Kling 1.6 [Kuaishou Technology, 2025]	22.4242	0.6289	0.6662	0.1353	0.9311	0.9504	0.1662	0.0064	**0.7227**	13.6%
Kling O3 [Kuaishou Technology, 2026]	32.0901	2.4129	0.7284	0.1933	0.9089	0.9863	0.2040	0.0094	0.5577	–
Ours	**12.8589**	**0.1404**	**0.8742**	**0.0615**	**0.9528**	0.9881	**0.0468**	0.0047	0.7193	**79.4%**

1.6 benefits from region-level hints but still trails in garment fidelity and reconstruction quality, and Kling O3 tends to regenerate rather than locally edit the source video.

4.3 Human and Reward Evaluation

Human preference study. We use a four-method shortlist consisting of Ours, MagicTryOn, UniVideo, and Kling 1.6. We sample 50 TripVVT-Bench examples and assign them to 20 participants, yielding 1,000 votes in total. For each example, participants view the anonymized outputs in randomized order and rank the four results according to overall video quality, reference-garment fidelity, source preservation, background consistency, and temporal coherence. We count the first-place votes for each method and report the resulting first-place vote ratio in Table 2.

Reward-human agreement. We further evaluate whether the proposed reward reflects human preference. We sample 100 TripVVT-Bench test examples, generate four videos for each example, and ask human annotators to rank the results according to overall try-on quality. The reward model scores the same videos with the fixed try-on rubric, and we compare the reward-induced ranking with human ranking using pairwise agreement and rank-correlation metrics. As shown in Table 3, the proposed MLLM reward has positive agreement with human preference, suggesting that it provides a meaningful training signal for try-on-specific post-training. Appendix C.3 defines the agreement metrics and gives additional reward-model details.

4.4 Ablation Studies

We conduct ablations on TripVVT-Bench to verify the roles of MLLM conditioning, garment-aware reasoning, source anchoring, generator-side garment tokens, and DiffusionNFT post-training. The variants replace the MLLM with the original T5 encoder, remove the reference garment from the MLLM input, remove the source-video condition, remove generator-side reference-garment tokens, or use only the supervised model without post-training.

Table 4 shows that all components contribute to the final model. Replacing the MLLM with T5 causes the largest drop, while the two garment pathways are complementary: the MLLM garment input supports high-level alignment, and garment tokens preserve fine-grained appearance. Source-video conditioning improves preservation and temporal stability, and DiffusionNFT post-training improves both MLLM-Reward and independent quality metrics.

Table 3: Reward-human agreement on TripVVT-Bench. Agreement is computed on 100 test samples with four generated videos per sample.

Metric	Pairwise agreement ↑	Spearman ρ ↑	Kendall τ ↑
Ours MLLM reward	65.5%	0.368	0.310

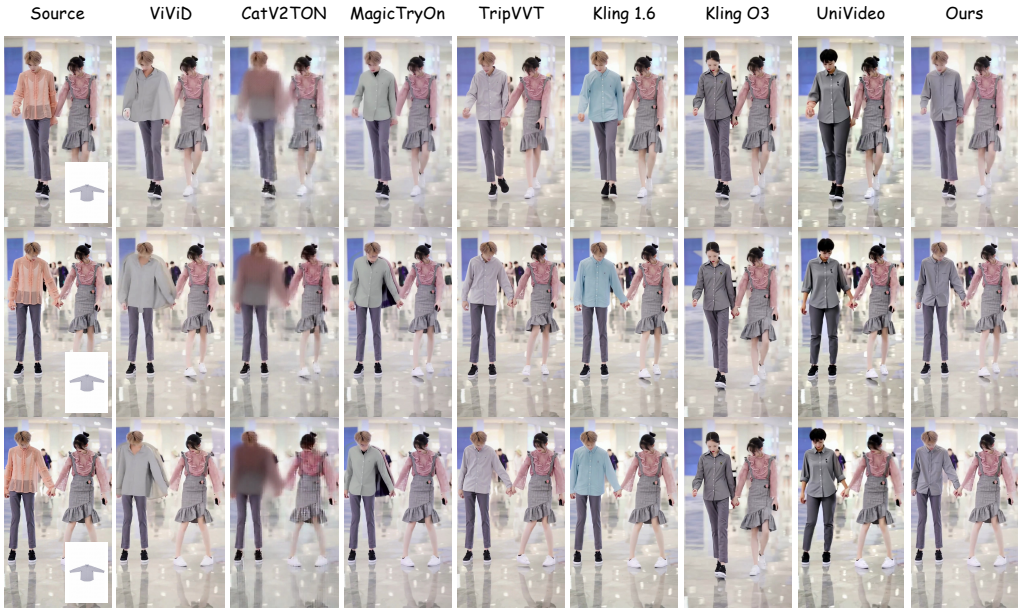


Figure 3: Qualitative results on TripVVT-Bench. Our method preserves the source video structure and transfers the reference garment to the instructed subject with better temporal consistency.

Table 4: Ablation studies on TripVVT-Bench. We use the same metric setting as the TripVVT-Bench quantitative comparison. Best results are shown in **bold**, and second-best results are underlined; ties share the same rank, with the next distinct value marked as second-best when applicable.

Variant	VFID_I ↓	VFID_R ↓	SSIM ↑	LPIPS ↓	CLIP-I ↑	CLIP-F ↑	BG-L1 ↓	BG-DINO ↓	MLLM-Reward ↑
w/o MLLM, using T5	18.3755	0.3496	0.8467	0.0808	0.9320	**0.9882**	0.0584	0.0053	0.6581
w/o garment input to MLLM	13.0479	**0.1161**	0.8717	0.0633	0.9514	0.9881	0.0487	0.0048	0.7064
w/o source-video condition	14.5686	0.1938	0.8659	0.0673	0.9476	0.9878	0.0538	0.0049	0.7068
w/o reference-garment tokens	14.8048	0.1996	0.8681	0.0675	0.9438	0.9881	0.0520	0.0048	0.6321
w/o DiffusionNFT post-training	13.8390	0.1381	0.8688	0.0656	0.9482	**0.9882**	0.0490	0.0048	0.6859
Full model	**12.8589**	0.1404	**0.8742**	**0.0615**	**0.9528**	0.9881	**0.0468**	**0.0047**	**0.7193**

4.5 Qualitative Results and Generalization

Results on TripVVT-Bench. Fig. 3 shows that, on TripVVT-Bench, our method better preserves source structure and non-target regions while transferring the reference garment, especially under camera motion, occlusion, and multi-person ambiguity. It also produces more reference-faithful and temporally stable garment details than general video editors.

Results on ViViD-S. Additional ViViD-S examples in Appendix I, Figs. 7 and 9, further show stable source preservation and reference-garment transfer on controlled indoor videos, complementing Table 1.

5 Conclusion

We presented InstructVVT, an instruction-driven framework for reference-based video virtual try-on that uses only a source video, a reference garment, and a natural-language instruction at inference time. Instead of relying on masks, poses, parsing maps, DensePose-like representations, garment lines, or other auxiliary spatial priors, the framework decomposes try-on control into three com-

plementary signals: MLLM edit tokens for target-aware and garment-aware intent understanding, source-video latents for structure and motion anchoring, and generator-side garment tokens for fine-grained reference appearance preservation. We further introduced a try-on-specific MLLM reward and DiffusionNFT-based post-training to align the generator with garment fidelity, target correctness, source preservation, motion naturalness, and temporal consistency. Experiments on multiple video try-on benchmarks, together with systematic ablations, show that this design provides a practical and effective path toward controllable in-the-wild video try-on without inference-time auxiliary structural inputs.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (Grant No. 62406134, Grant No. U24A20330), Jiangsu Provincial Science & Technology Major Project (Grant No. BG2024042), the Suzhou Key Technologies Project (Grant No. SYG2024136) and the Nanjing University-China Mobile Communications Group Co. Ltd. Joint Institute.

References

Qingyan Bai, Qiuyu Wang, Hao Ouyang, Yue Yu, Hanlin Wang, Wen Wang, Ka Leong Cheng, Shuailei Ma, Yanhong Zeng, Zichen Liu, Yinghao Xu, Yujun Shen, and Qifeng Chen. Scaling instruction-based video editing with a high-quality synthetic dataset. *arXiv preprint arXiv:2510.15742*, 2025a. doi: 10.48550/arXiv.2510.15742.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint*, arXiv:2511.21631, 2025b. doi: 10.48550/arXiv.2511.21631. URL `https://arxiv.org/abs/2511.21631`.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, 2024.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. InstructPix2Pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4733, 2017. doi: 10.1109/CVPR.2017.502.

Tianyu Chang, Xiaohao Chen, Zhichao Wei, Xuanpu Zhang, Qingguo Chen, Weihua Luo, Peipei Song, and Xun Yang. PEMF-VTO: Point-enhanced video virtual try-on via mask-free paradigm. *arXiv preprint*, arXiv:2412.03021, 2024. doi: 10.48550/arXiv.2412.03021.

Seunghwan Choi, Sunghyun Park, Minsoo Lee, and Jaegul Choo. VITON-HD: high-resolution virtual try-on via misalignment-aware normalization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 14131–14140, 2021. doi: 10.1109/CVPR46437.2021.01391.

Yisol Choi, Sangkyung Kwak, Kyungmin Lee, Hyungwon Choi, and Jinwoo Shin. Improving diffusion models for authentic virtual try-on in the wild. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXXVI*, volume 15144 of *Lecture Notes in Computer Science*, pages 206–235, 2024. doi: 10.1007/978-3-031-73016-0_13.

Zheng Chong, Xiao Dong, Haoxiang Li, Shiyue Zhang, Wenqing Zhang, Xujie Zhang, Hanqing Zhao, and Xiaodan Liang. CatVTON: Concatenation is all you need for virtual try-on with diffusion models, 2024. URL `https://arxiv.org/abs/2407.15886`.

Zheng Chong, Wenqing Zhang, Shiyue Zhang, Jun Zheng, Xiao Dong, Haoxiang Li, Yiling Wu, Dongmei Jiang, and Xiaodan Liang. CatV2TON: Taming diffusion transformers for vision-based virtual try-on with temporal concatenation. *arXiv preprint*, arXiv:2501.11325, 2025. doi: 10.48550/ARXIV.2501.11325.

DecartAI Team. Lucy Edit: Open-weight text-guided video editing. Technical report, 2025. URL `https://huggingface.co/decart-ai/Lucy-Edit-Dev`.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. DPOK: Reinforcement learning for fine-tuning text-to-image diffusion models. In *Advances in Neural Information Processing Systems*, 2023.

Zixun Fang, Wei Zhai, Aimin Su, Hongliang Song, Kai Zhu, Mao Wang, Yu Chen, Zhiheng Liu, Yang Cao, and Zheng-Jun Zha. ViViD: Video virtual try-on using diffusion models. *arXiv preprint*, arXiv:2405.11794, 2024. doi: 10.48550/ARXIV.2405.11794.

Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. Guiding instruction-based image editing via multimodal large language models. In *International Conference on Learning Representations*, 2024.

Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. DensePose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7297–7306, 2018.

Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018. doi: 10.1109/CVPR.2018.00685.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Glenn Jocher, Jing Qiu, and Ayush Chaurasia. Ultralytics YOLO, 2023. URL `https://github.com/ultralytics/ultralytics`. Version 8.0.0.

Rawal Khirodkar, Timur M. Bagautdinov, Julieta Martinez, Su Zhaoen, Austin James, Peter Selednik, Stuart Anderson, and Shunsuke Saito. Sapiens: Foundation for human vision models. In *European Conference on Computer Vision*, pages 206–228, 2024. doi: 10.1007/978-3-031-73235-5_12.

Jeongho Kim, Gyojung Gu, Minho Park, Sunghyun Park, and Jaegul Choo. StableVITON: Learning semantic correspondence with latent diffusion model for virtual try-on. *arXiv preprint*, arXiv:2312.01725, 2023. doi: 10.48550/ARXIV.2312.01725.

Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. VIEScore: Towards explainable metrics for conditional image synthesis evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 12268–12290, 2024a. doi: 10.18653/V1/2024.ACL-LONG.663.

Max Ku, Cong Wei, Weiming Ren, Harry Yang, and Wenhu Chen. AnyV2V: A plug-and-play framework for any video-to-video editing tasks. *arXiv preprint arXiv:2403.14468*, 2024b.

Kuaishou Technology. Kuaishou Kling AI unveils "multi-image reference" feature to further tackle AI video consistency challenges. `https://ir.kuaishou.com/news-releases/news-release-details/kuaishou-kling-ai-unveils-multi-image-reference-feature-further`, 2025. Mentions the December 2024 launch of Kling AI 1.6. Accessed: 2026-05-05.

Kuaishou Technology. Kling AI launches 3.0 model, ushering in an era where everyone can be a director. https://ir.kuaishou.com/news-releases/news-release-details/kling-ai-launches-30-model-ushering-era-where-everyone-can-be, 2026. Accessed: 2026-05-05.

Guangyuan Li, Siming Zheng, Hao Zhang, Jinwei Chen, Junsheng Luan, Binkai Ou, Lei Zhao, Bo Li, and Peng-Tao Jiang. MagicTryOn: Harnessing diffusion transformer for garment-preserving video virtual try-on. *arXiv preprint*, arXiv:2505.21325, 2025a. doi: 10.48550/ARXIV.2505.21325.

Zongjian Li, Zheyuan Liu, Qihui Zhang, Bin Lin, Feize Wu, Shenghai Yuan, Zhiyuan Yan, Yang Ye, Wangbo Yu, Yuwei Niu, Shaodong Wang, Xinhua Cheng, and Li Yuan. Uniworld-V2: Reinforce image editing with diffusion negative-aware finetuning and MLLM implicit feedback. *arXiv preprint*, arXiv:2510.16888, 2025b. doi: 10.48550/arXiv.2510.16888.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Eyal Molad, Eliahu Horwitz, Dani Valevski, Alex Rav-Acha, Yossi Matias, Yael Pritch, Yaniv Leviathan, and Yedid Hoshen. Dreamix: Video diffusion models are general video editors. *arXiv preprint arXiv:2302.01329*, 2023.

Davide Morelli, Matteo Fincato, Marcella Cornia, Federico Landi, Fabio Cesari, and Rita Cucchiara. Dress code: High-resolution multi-category virtual try-on. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2022, New Orleans, LA, USA, June 19-20, 2022*, pages 2230–2234, 2022. doi: 10.1109/CVPRW56347.2022.00243.

Chong Mou, Qichao Sun, Yanze Wu, Pengze Zhang, Xinghui Li, Fulong Ye, Songtao Zhao, and Qian He. InstructX: Towards unified visual editing with MLLM guidance. *arXiv preprint*, arXiv:2510.08485, 2025. doi: 10.48550/arXiv.2510.08485.

Xichen Pan, Satya Narayan Shukla, Aashu Singh, Zhuokai Zhao, Shlok Kumar Mishra, Jialiang Wang, Zhiyang Xu, Jiuhai Chen, Kunpeng Li, Felix Juefei-Xu, Ji Hou, and Saining Xie. Transfer between modalities with MetaQueries. *arXiv preprint*, arXiv:2504.06256, 2025. doi: 10.48550/arXiv.2504.06256.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4172–4182, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763, 2021.

Runway. Runway Aleph: A new way to edit, transform and generate video, 2025. URL https://runway.com/research/introducing-runway-aleph.

Dingbao Shao, Song Wu, Shenyi Wang, Ye Wang, Ziheng Tang, Fei Liu, Jiang Lin, Xinyu Chen, Qian Wang, Ying Tai, Jian Yang, and Zili Yi. TripVVT: A large-scale triplet dataset and a coarse-mask baseline for in-the-wild video virtual try-on. arXiv preprint arXiv:2604.27958, 2026.

Le Shen, Yanting Kang, Rong Huang, and Zhijie Wang. MFP-VTON: Enhancing mask-free person-to-person virtual try-on via diffusion transformer. *arXiv preprint*, arXiv:2502.01626, 2025. doi: 10.48550/arXiv.2502.01626.

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. doi: 10.48550/arXiv.2503.20314. URL https://arxiv.org/abs/2503.20314.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric and challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8228–8238, 2024.

Zhenchen Wan, Yanwu Xu, Dongting Hu, Weilun Cheng, Tianxi Chen, Zhaoqing Wang, Feng Liu, Tongliang Liu, and Mingming Gong. MF-VITON: High-fidelity mask-free virtual try-on with minimal input. *arXiv preprint*, arXiv:2503.08650, 2025. doi: 10.48550/arXiv.2503.08650.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

Cong Wei, Quande Liu, Zixuan Ye, Qiulin Wang, Xintao Wang, Pengfei Wan, Kun Gai, and Wenhu Chen. UniVideo: Unified understanding, generation, and editing for videos. *arXiv preprint*, arXiv:2510.08377, 2025. doi: 10.48550/arXiv.2510.08377.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. MagicBrush: A manually annotated dataset for instruction-guided image editing. In *Advances in Neural Information Processing Systems*, 2023.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. doi: 10.1109/CVPR.2018.00068.

Xuanpu Zhang, Dan Song, Pengxin Zhan, Tianyu Chang, Jianhao Zeng, Qingguo Chen, Weihua Luo, and An-An Liu. BooW-VTON: Boosting in-the-wild virtual try-on via mask-free pseudo data training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26399–26408, 2025.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. DiffusionNFT: Online diffusion reinforcement with forward process. *arXiv preprint*, arXiv:2509.16117, 2025. doi: 10.48550/arXiv.2509.16117.

Luyang Zhu, Dawei Yang, Tyler Zhu, Fitsum Reda, William Chan, Chitwan Saharia, Mohammad Norouzi, and Ira Kemelmacher-Shlizerman. TryOnDiffusion: A tale of two UNets. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 4606–4615, 2023. doi: 10.1109/CVPR52729.2023.00447.

Tongchun Zuo, Zaiyu Huang, Shuliang Ning, Ente Lin, Chao Liang, Zerong Zheng, Jianwen Jiang, Yuan Zhang, Mingyuan Gao, and Xin Dong. DreamVVT: Mastering realistic video virtual try-on in the wild via a stage-wise diffusion transformer framework. *arXiv preprint*, arXiv:2508.02807, 2025. doi: 10.48550/ARXIV.2508.02807.

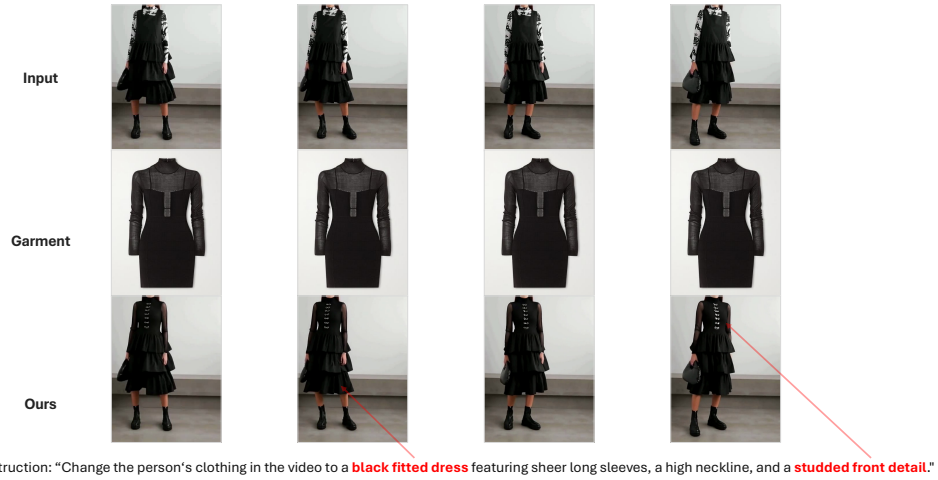


Figure 4: A limitation case under conflict between text semantics and the reference garment image. When the instruction describes garment attributes that are inconsistent with the visual reference, the model may not fully preserve the reference garment appearance.

A Limitations

The current design still depends on the visual reasoning ability of the MLLM and the quality of the reward model. Extremely long videos, severe occlusion, unusual garment geometry, and instructions requiring large physical changes may remain challenging. We also observe a failure mode when the text instruction semantically conflicts with the reference garment image. For example, if the instruction describes a garment attribute that is inconsistent with the reference image, the model may compromise between the text semantics and the visual reference, and the generated clothing may not fully follow the reference garment, as illustrated in Fig. 4.

Future work can improve conflict resolution between language and visual garment evidence, strengthen long-horizon consistency, expand reward supervision, and study more interactive forms of user control.

B Architecture details

Architecture-specific settings. We uniformly sample four source frames for MLLM conditioning. The selected query-token hidden states are projected to the DiT cross-attention dimension by a two-layer MLP connector with GELU activation. The garment patch embedding is randomly initialized and trained from scratch. The resulting garment tokens participate in DiT self-attention as conditioning tokens, but are not decoded as output tokens. During DiffusionNFT post-training, the conditioning MLLM is frozen and only the video diffusion generator is updated; the reward MLLM is a separate frozen model.

Inference. At inference time, the model samples four source frames and feeds them, together with the reference garment and instruction, to the conditioning MLLM to obtain edit tokens. The full source video is encoded into source latents, and the reference garment is encoded into garment tokens. Starting from Gaussian noise, the DiT denoises the target-video latent conditioned on source latents, garment tokens, and MLLM edit tokens. The final latent is decoded by the VAE into the edited video.

C Training details

C.1 Training data construction

Our training corpus consists of three parts: 100K image triplets, 10K ViViD video triplets, and the public TripVVT-10K training set [Shao et al., 2026]. TripVVT-10K is used as released. The other two parts are internally constructed supplementary triplets following the same source–reference–target format, where each sample contains an original person image or video, a reference garment, and a synthesized try-on target.

Instruction annotation. Each training triplet is paired with a natural-language try-on instruction. For the generated image and ViViD triplets, we use Qwen3-VL-235B-A22B-Instruct to describe the intended garment transfer for each source–garment pair. To avoid a narrow instruction distribution, we create two instruction styles for every pair: a concise command and a more descriptive request that includes additional visual context. During training, one of the two instructions is sampled with equal probability. For TripVVT-10K, where a video may contain multiple people, we first use the provided human mask to indicate the target person and then feed the marked target together with the garment reference to Qwen3-VL-235B-A22B-Instruct, so that the resulting instruction refers to the intended person rather than to the scene in general.

Image triplets. For image training data, we build candidate person–garment pairs from public image try-on datasets. Person images are taken from VITON-HD [Choi et al., 2021], and reference garments are sampled from DressCode [Morelli et al., 2022]. To increase garment diversity, each person's image is paired with multiple randomly selected garments, with fewer random pairings for dress samples to avoid category imbalance. We then run CatVTON [Chong et al., 2024] on each candidate pair to synthesize the target try-on image, resulting in candidate triplets of the form $\langle$person image, garment image, synthesized try-on image$\rangle$.

The candidate image triplets are filtered in two stages. First, we remove samples that introduce large pose changes after synthesis. We detect 17 human keypoints in both the original and synthesized images using Sapiens [Khirodkar et al., 2024], keep keypoints with confidence above 0.3 in both images, and compute the mean normalized keypoint displacement. This displacement is converted into a pose similarity score, and only candidates with a similarity above 0.95 are kept. Second, we use a VIEScore-style evaluator [Ku et al., 2024a] adapted to try-on data filtering. The evaluator scores perceptual quality and semantic consistency, covering clothing naturalness, artifact severity, identity preservation, and garment-transfer correctness. We combine the two scores with a harmonic mean and keep samples whose overall score is above 0.9. From the filtered pool, we retain 100K image triplets for training.

ViViD video triplets. For supplementary video data, we construct triplets from ViViD [Fang et al., 2024]. Because raw garment masks in video try-on data can be temporally unstable, we first filter source clips before generating synthetic targets. For each mask sequence, we compute frame-wise mask area, centroid, bounding box, normalized area change, inter-frame IoU, and centroid displacement. Frames with abrupt area changes, low temporal overlap, or large spatial jumps are marked as unstable. We also use a YOLO-based human pose detector [Jocher et al., 2023] to check whether the mask lies in a plausible body region for the corresponding garment type, e.g., upper-body, lower-body, or dress.

Stable intervals are obtained by aggregating the temporal and anatomical checks and extracting continuous segments without detected anomalies. We keep segments that satisfy a minimum temporal length requirement and split overly long intervals when needed. Each retained person sequence is paired with a target garment, and MagicTryOn [Li et al., 2025a] is used to synthesize the corresponding try-on video. We do not apply an additional generated-video filtering stage beyond the source-mask stability filtering. The final supplementary video set contains 10K ViViD triplets.

C.2 Stage-wise training settings

We summarize the reproducibility-critical protocol in prose and reserve tables for quantities that are easier to compare visually. The reported results use two public video try-on benchmarks. ViViD-S contains 180 videos released by CatV2TON, and TripVVT-Bench contains 100 videos released

Table 5: Evaluation benchmarks and inference settings.

Benchmark	Videos	Frames	Resolution	Instruction source
ViViD-S	180	61	832×624	MLLM-generated
TripVVT-Bench	100	49	832×480	MLLM-generated

Table 6: Training and compute settings.

Stage	Objective	Data	Trainable	Steps	Batch	EMA	Hardware
1	Interface alignment	Mixed	Q, LoRA, Conn	20K	1	–	8 H100
2	Full-condition SFT	Mixed	MLLM, DiT	25K	1	–	8 H100
3	High-resolution SFT	TripVVT-10K	MLLM, DiT	15K	1	0.999	8 H100
RL	DiffusionNFT	Rollouts	Gen	180	30	0.5	40 H100

by TripVVT. For both benchmarks, the evaluation instructions are generated by an MLLM. Auxiliary annotations required by condition-based baselines are taken from the benchmark release when available; otherwise, we follow the corresponding method's official preprocessing pipeline.

All open-source baselines are evaluated with their official checkpoints. For baselines that require masks, poses, or related structural inputs, we use their official condition-generation pipelines. Kling 1.6 is evaluated through the official web interface with manually selected edit points, matching its intended point-based interaction mode. The generated results, per-sample instructions, evaluation sample lists, metric scripts, and baseline configuration files are the key artifacts needed to reproduce the reported numbers; their public release is subject to the code/data review process described in the checklist.

We train the model in four stages. The first three stages are supervised fine-tuning stages with batch size 1, and the last stage is DiffusionNFT-based RL post-training with batch size 30. All stages use AdamW with learning rate 1×10^{-5}. EMA is used only in Stage 3 and RL, with decay values 0.999 and 0.5, respectively. For the Qwen3-VL-8B conditioning MLLM in InstructVVT, the LoRA rank is 16, and the numbers of image and video query tokens are 256 and 512.

For RL post-training, the 40 H100 GPUs are organized as five 8-GPU machines. On each machine, two GPUs serve the reward model and six GPUs train the policy model. Instruction annotation and reward scoring are also run on H100 GPUs. We omit absolute paths, cache locations, service ports, timeout values, and dataloader internals because they are engineering details rather than reproducibility-defining experimental settings.

C.3 Reward model details

The reward model is a frozen Qwen3-VL-32B that is separate from the Qwen3-VL-8B conditioning MLLM. It is used only during DiffusionNFT post-training and for reporting MLLM-Reward. For each generated sample, the reward model receives the reference garment image, the original source video, the generated try-on video, and the instruction. The reward MLLM processes source and generated videos at 2 FPS. It is prompted with a fixed try-on rubric, shown in Fig. 6, and asked to directly output a single score from 0 to 5.

For the reward-human agreement study in Sec. 4.3, each test example has four generated videos and therefore six pairwise comparisons. Pairwise agreement is the fraction of video pairs for which the reward-induced preference agrees with the human ranking. Spearman's ρ measures rank correlation between the reward ranking and the human ranking, while Kendall's τ measures agreement in terms of concordant and discordant ordered pairs. These metrics are used only to validate the reward against human preference; the post-training objective uses the scalar score-token reward described below.

We use a non-CoT score-token reward. Instead of parsing the generated text, we compute the expected score from the first-token logits of the six score tokens $\{0, 1, 2, 3, 4, 5\}$ and normalize it to $[0, 1]$, as described in Eq. 7. This provides a smooth scalar reward and avoids brittle text parsing.

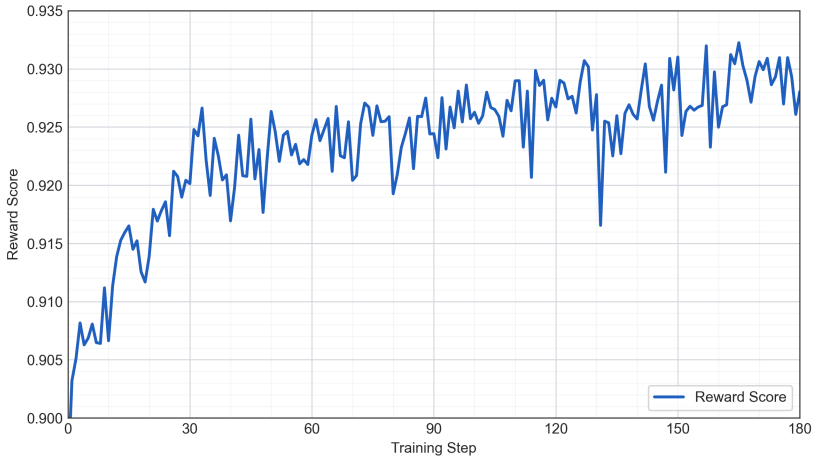


Figure 5: Reward trajectory during DiffusionNFT RL post-training. The curve tracks the MLLM reward used to optimize the video generator in the RL stage.

C.4 Other implementation details

We use a fixed negative prompt for Wan-style video sampling and keep it identical across supervised and RL stages. Unless otherwise specified, all RL experiments use random seed 42.

D Human evaluation details

The human preference study uses 50 TripVVT-Bench examples and four methods: Ours, MagicTryOn, UniVideo, and Kling 1.6. Each evaluation task shows the source video, the reference garment, the instruction, and four anonymized generated videos in randomized order. Participants are asked to rank the four results according to overall video try-on quality, considering whether the instructed target clothing is replaced by the reference garment, whether the source identity, body shape, motion, and background are preserved, whether the garment appearance is temporally stable, and whether the video is visually natural. We collect only preference rankings and do not collect personally identifying information from participants.

The instruction shown to participants is:

> You will see a source video, a reference garment, an editing instruction, and four anonymized generated videos. Please rank the four generated videos from best to worst according to overall video virtual try-on quality. A better result should apply the reference garment to the instructed target person, preserve non-edited identity, pose, motion, and background content, maintain temporal consistency, and avoid visual artifacts.

The reward-human agreement study follows the same ranking interface and criteria, but is used to compare human rankings with reward-induced rankings rather than to compute the four-method first-place vote ratios. Participants are informed that the study evaluates generated video try-on results for research purposes and can stop the evaluation at any time. Compensation is handled under the authors' internal annotation policy and is independent of the submitted rankings.

E Additional analysis of metric discrepancies

ViViD-S metric behavior. The ViViD-S results reveal an important limitation of generic video quality metrics for try-on evaluation. MagicTryOn performs strongly on VFID, CLIP-F, and background metrics, outperforming ViViD and CatV2TON on several low-level or distribution-level measures. However, its MLLM-Reward is substantially lower than those of ViViD, CatV2TON, and TripVVT. Qualitative inspection suggests that ViViD-S is mostly composed of relatively sim-

System prompt for video try-on reward scoring.

Human: You are evaluating a video virtual try-on result.

You will receive:
1. A reference garment image.
2. The original/source video.
3. The generated try-on video.

Instruction: {`prompt`}
Requirements: {`requirement`}

Rate the generated try-on video from 0 to 5 based on the following criteria.

(1) Target Clothing Replacement
– The clothing on the specified target body region should be replaced with the reference garment.
– If the model edits the wrong person, wrong region, or does not apply the garment, the score should be low.

(2) Reference-Garment Fidelity
– The replaced clothing should match the reference garment in category, color, texture, pattern, shape, and visible design details.
– Major mismatch with the reference garment should be penalized.

(3) Source Preservation
– Human regions outside the edited clothing area should be preserved, including identity, pose, body shape, face, hands, skin, and background.
– Unnecessary changes to non-edited regions should be penalized.

(4) Visual Quality and Temporal Stability
– The generated video should be visually natural, temporally stable, and free of severe artifacts.
– Flickering clothing, inconsistent details, or severe distortions should be penalized.

Use the following scale:

0: The try-on fails, edits the wrong target region, ignores the reference garment, or has severe artifacts.
5: The specified clothing region matches the reference garment accurately, all non-edited human regions and the background are preserved, and the video is visually high quality.

Response Format: directly output one score number from 0 to 5.

Do not output anything else.

Figure 6: System prompt used for MLLM reward scoring.

ple indoor videos, where many methods can preserve the person and background reasonably well; under this setting, a small number of severe target-clothing replacement failures can dominate try-on-specific scoring, because the reward penalizes wrong or incomplete garment transfer more strongly than generic video metrics.

UniVideo on ViViD-S. UniVideo shows a different behavior. As a general instruction-guided video editor, it achieves very high CLIP-F and a high MLLM-Reward, suggesting that it can often understand and execute the requested edit. Nevertheless, its VFID_I, VFID_R, BG-L1, and BG-DINO are much worse than specialized try-on methods. This indicates that general video editing ability alone is insufficient for video virtual try-on: the model may follow the instruction, but still fail to preserve the source distribution, background, and local editing constraints required by high-quality try-on.

Commercial editors on TripVVT-Bench. The comparison with commercial editors further highlights the difficulty of video virtual try-on. Kling 1.6 obtains a relatively strong MLLM-Reward among non-specialized models, which is consistent with its point/region-based editing interface: the additional region-level hint provides useful localization information, similar in spirit to the auxiliary spatial priors used by previous try-on systems. However, it still lags behind InstructVVT in garment fidelity, reconstruction quality, and human preference. Kling O3, despite being a more recent general-purpose multimodal video generation system, performs substantially worse on this benchmark. Qualitative inspection suggests that Kling O3 tends to regenerate the video according to the input reference and instruction rather than strictly preserving the source layout and performing a localized clothing edit. These results show that even strong general video generation models do not automatically solve video virtual try-on, and that recovering try-on-specific controllability without inference-time structural priors remains necessary.

F Additional ablation analysis

The ablation variants are designed to isolate the major condition pathways in InstructVVT. Replacing MLLM conditioning with the original T5 text encoder tests whether MLLM-based conditioning provides useful video- and garment-aware editing context beyond text-only conditioning. Removing the reference garment from the MLLM input while keeping the generator-side garment tokens isolates the contribution of giving the MLLM access to the reference garment for garment-to-person alignment. Removing the source-video condition tests whether the input video latent is necessary for preserving motion, layout, and non-edited regions. Removing generator-side reference-garment tokens by zeroing the reference-garment latent tests whether the garment token branch contributes fine-grained reference appearance beyond the reference information already available to the MLLM. Finally, removing DiffusionNFT-based post-training tests whether MLLM reward post-training improves try-on-specific qualities beyond supervised fine-tuning.

The results in Table 4 support this decomposition. Replacing the MLLM with T5 leads to worse performance, especially on metrics related to target correctness, source preservation, instruction following, and garment-reference alignment. Removing the garment input from the MLLM weakens garment-reference alignment while retaining the generator-side garment tokens, indicating that access to the reference garment helps the MLLM provide high-level guidance on how the garment should be applied to the target person. Removing the source-video condition harms preservation and temporal stability, confirming that source latents are important anchors for the original motion, layout, and background. When generator-side reference-garment tokens are removed, the MLLM can still provide garment-aware intent, but reference-specific fidelity and fine details degrade. Removing DiffusionNFT-based post-training also degrades performance, indicating that the proposed MLLM reward provides useful preference feedback for improving garment fidelity, preservation, and temporal consistency. The full model improves the reward score and most independent metrics, suggesting that the gain reflects broader try-on quality improvement rather than only reward-score optimization.

G Broader impacts and safeguards

Instruction-driven video virtual try-on can support online retail, digital content creation, and accessibility by reducing the need for repeated physical try-on or manual video editing. At the same time, the same capability may be misused to alter a person's appearance without consent, create misleading fashion or identity-related media, or generate edited videos that viewers may mistake for authentic recordings. These risks are especially relevant for in-the-wild videos that contain identifiable people.

Responsible deployment should therefore require consent from depicted individuals, restrict use on sensitive or non-consensual imagery, and clearly disclose generated or edited content. If models, demos, or generated data are released, we plan to include usage restrictions, safety filtering for inappropriate inputs, and documentation of intended and prohibited uses. We also encourage downstream systems to combine try-on generation with provenance or watermarking mechanisms where feasible.

H Asset licenses and terms

We use external assets for data construction, benchmark evaluation, baseline comparison, filtering, annotation, reward modeling, and generator initialization. Table 7 lists only the asset category, role, and public license or terms status. When a release does not specify a standard open-source license, we report the official terms status rather than inferring one.

Assets governed by non-commercial licenses, academic-use agreements, or service terms are used only within the corresponding research, evaluation, or service-access context. Any public release of derived data, generated results, checkpoints, or code will be reviewed against the applicable asset terms.

Additional qualitative comparisons on TripVVT-Bench are shown in Fig. 10. These examples supplement the main-paper visual comparison by covering more in-the-wild scenes, reference garments, and target-person configurations.

Table 7: External assets used in this work and their public license or terms status.

Asset	Category	Role	License / terms
VITON-HD	Dataset	Image source persons	CC BY-NC 4.0
DressCode	Dataset	Reference garments	Academic non-commercial terms
ViViD	Dataset / benchmark	Video triplets; ViViD-S	Apache-2.0
TripVVT-10K / TripVVT-Bench	Dataset / benchmark	Training/evaluation	Official release terms
CatVTON	Method / baseline	Image target synthesis	CC BY-NC-SA 4.0
MagicTryOn	Method / baseline	Video target synthesis	CC BY-NC-SA 4.0
UniVideo	Method / baseline	General video editing baseline	MIT
Sapiens	Tool	Keypoint filtering	CC BY-NC 4.0; Apache-2.0 portions
Ultralytics YOLO	Tool	Pose/body-region checks	AGPL-3.0 or Enterprise
Qwen3-VL-8B / 32B / 235B-A22B-Instruct	Model	Condition; reward; annotation	Apache-2.0
Wan2.1-style backbone	Model	Generator initialization	Apache-2.0
Kling 1.6 / Kling O3	Service	Commercial baseline	Provider service terms

I Additional qualitative results

Additional qualitative comparisons are provided for both ViViD-S and TripVVT-Bench. ViViD-S visualizations complement Table 1 by showing frame-level garment transfer and preservation behavior on the controlled benchmark, while TripVVT-Bench visualizations cover more in-the-wild scenes and target-person configurations. We also include a qualitative ablation comparison to illustrate how the major conditioning pathways affect the generated videos.



Figure 7: Qualitative comparisons on ViViD-S. The examples show that InstructVVT transfers the reference garment while preserving the source video appearance and temporal structure, complementing the quantitative ViViD-S results in Table 1.

J Additional discussion of conditioning and evaluation

J.1 Scope of inference-time spatial-prior freedom

Spatial-prior freedom here refers to inference. The model receives the source video, reference garment image, and instruction, without an additional mask, pose, parsing map, DensePose representation, or garment-line input. This scope does not imply that spatial priors are absent from data preparation. As detailed in Appendix C.1, training-data construction uses CatVTON and MagicTryOn for target synthesis, Sapiens keypoints for image filtering, garment masks and pose-based body-region checks for video filtering, and the released human masks for target-person disambiguation during instruction annotation. These annotations support offline supervision construction; they are not required inputs to the deployed generator.

This distinction also helps position the method relative to mask-free try-on. Image-based methods such as BooW-VTON, MF-VITON, and MFP-VTON [Zhang et al., 2025, Wan et al., 2025, Shen et al., 2025] reduce inference-time spatial inputs while using synthesized supervision or mask-based models during data preparation or training. Their setting differs from maintaining reference fidelity

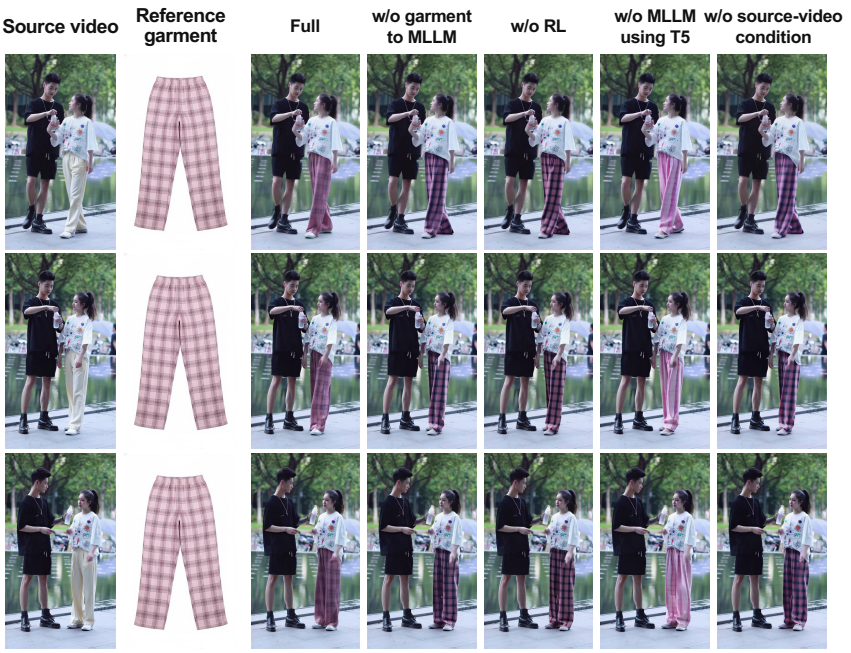


Figure 8: Qualitative ablation comparison on TripVVT-Bench. Replacing the MLLM with T5 weakens video- and garment-aware edit conditioning; removing the reference garment from the MLLM input reduces high-level garment-to-person alignment; removing source-video conditioning weakens preservation; and removing DiffusionNFT post-training reduces the overall try-on quality. These visual trends are consistent with the quantitative ablations in Table 4.

Table 8: Positioning by task and spatial guidance. "None" means no auxiliary spatial annotation is supplied at inference; source and reference images or videos are still required. "Text target" denotes language-based designation of the person and edit scope.

Method	Task	Spatial input at inference	Priors in training or data preparation	Text target
CatVTON (standard)	Image	Garment-region mask	Mask-based training	No
BooW-VTON	Image	None	Mask-based pseudo-data synthesis and localization loss	No
MF-VITON	Image	None	Mask-based teacher and pseudo-data synthesis	No
MFP-VTON	Image	None	Pseudo-data synthesis and garment-mask loss	No
PEMF-VTO	Video	Sparse point alignments	Mask-based pseudo-video synthesis	No
CatV2TON	Image / video	Masks and DensePose	Masks and DensePose	No
TripVVT	Video	Human mask, pose, garment lines	Spatial conditioning and data preparation	No
InstructVVT	Video	None	Offline synthesis, filtering, and annotation	Yes

and source preservation throughout a video. Among video methods, mask-free operation does not necessarily eliminate all spatial guidance: PEMF-VTO [Chang et al., 2024] uses point-based guidance, whereas CatV2TON and TripVVT [Chong et al., 2025, Shao et al., 2026] use explicit spatial conditions. InstructVVT instead uses the instruction to specify the target and edit scope, together with the source and garment representations described in Sec. 3.2.

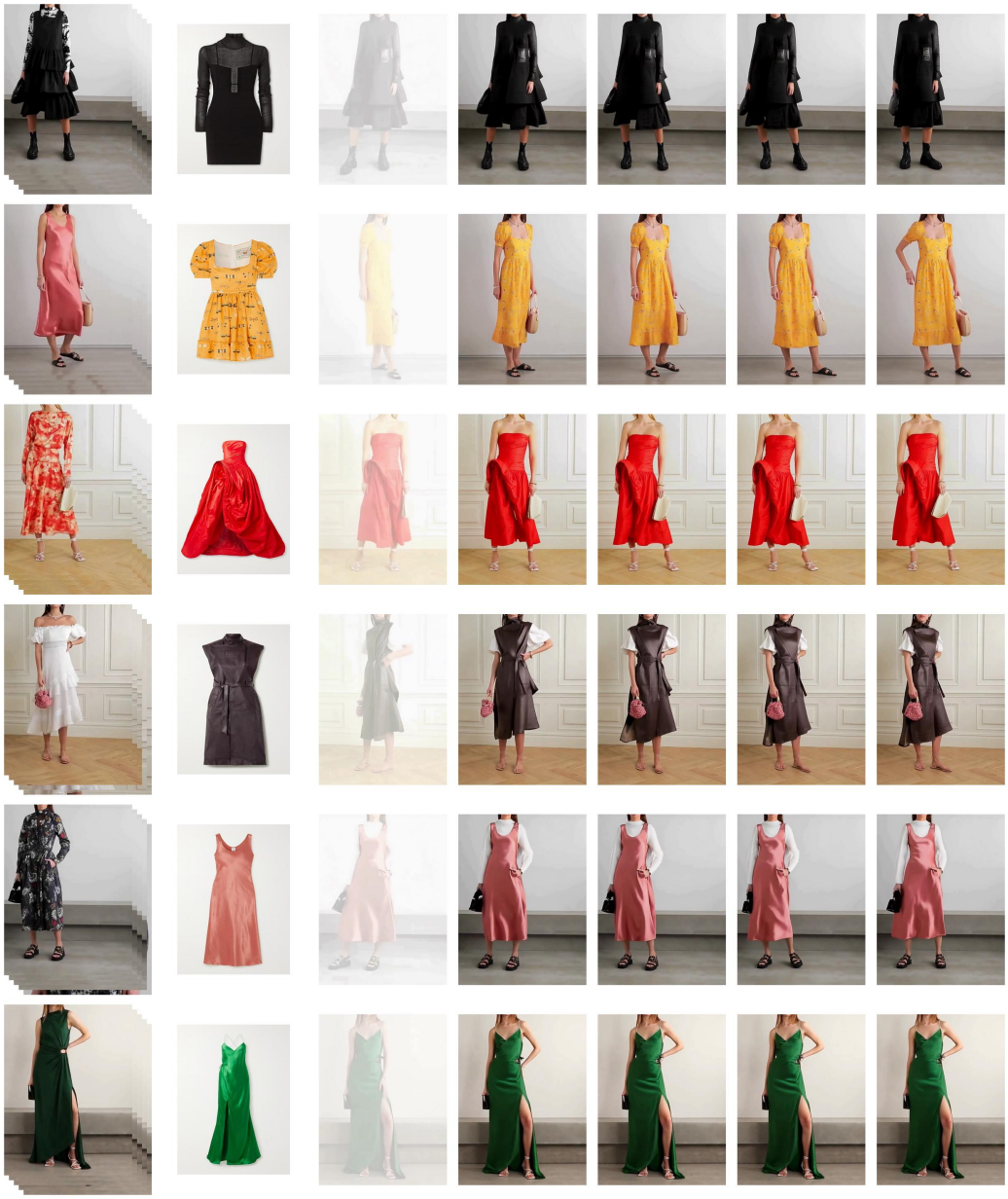


Figure 9: Additional ViViD-S visualizations. These samples provide a broader view of the method's behavior across different source videos and reference garments, showing stable garment appearance and source preservation across sampled frames.

Table 8 separates inference inputs from offline supervision. The CatVTON row refers to the standard mask-based configuration [Chong et al., 2024]; the authors also provide a separate mask-free release.[1] These versions should not be conflated when describing inference requirements. The comparison concerns the listed try-on configurations, not all possible extensions of each method.

[1] Official CatVTON mask-free release: `https://huggingface.co/zhengchong/CatVTON-MaskFree`.

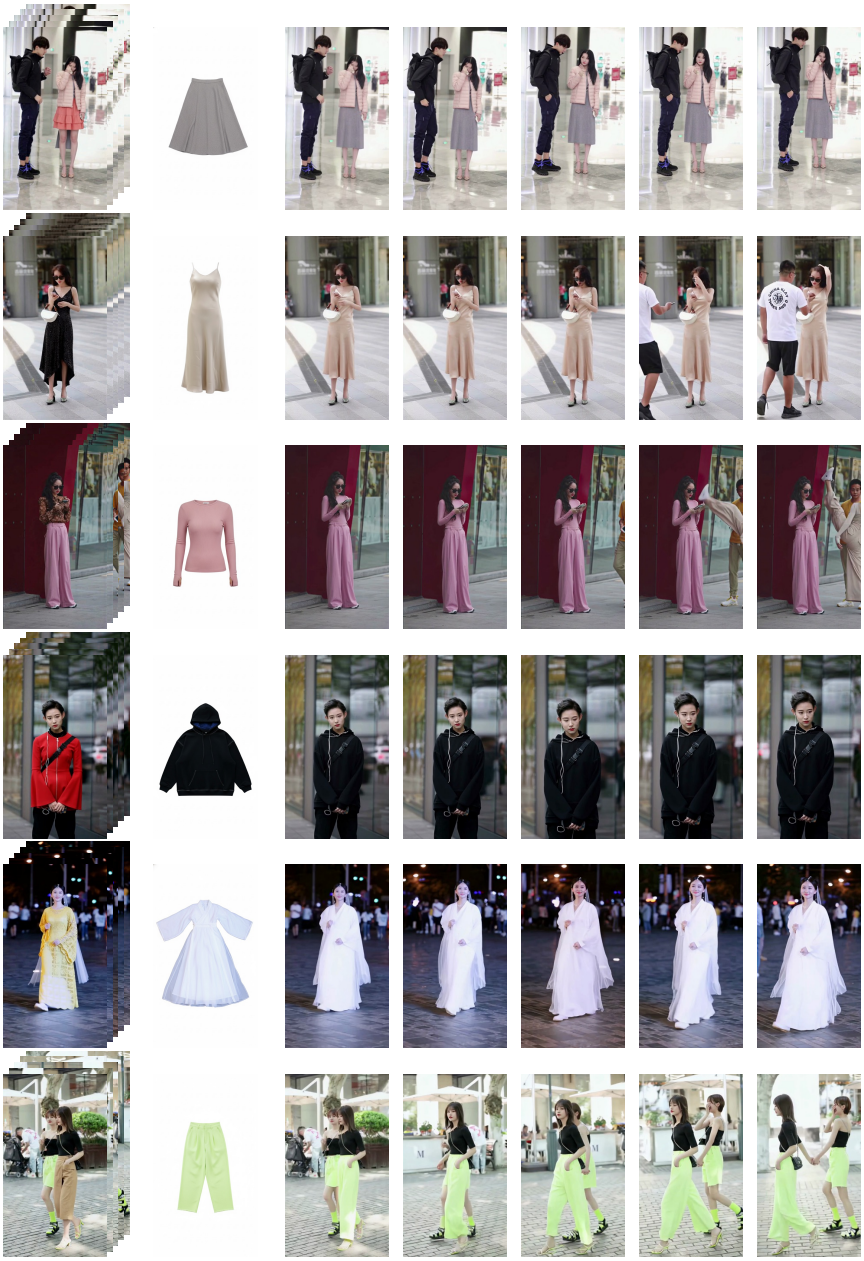


Figure 10: Additional qualitative comparisons on TripVVT-Bench. The examples cover diverse in-the-wild scenes, reference garments, and target-person configurations, complementing the main-paper qualitative results.

J.2 Interpreting the conditioning design and existing ablations

The learnable-query connection between an MLLM and a diffusion generator follows prior work [Pan et al., 2025, Mou et al., 2025]. The task-specific design here concerns how the source video and reference garment participate in this interface. The MLLM observes the garment together with the source frames and instruction, while an independent garment embedding provides visual tokens to the generator. Source latents are concatenated with noisy video latents along the channel dimension. Video and garment tokens then share the DiT token sequence, and the MLLM edit tokens provide cross-attention conditioning; only video-token positions are decoded. The corresponding ab-

lations in Table 4 examine these complementary pathways rather than claiming that each underlying architectural primitive is new.

In Tables 2 and 4, the T5 variant uses no auxiliary spatial priors and obtains lower VFID_I, LPIPS, and BG-L1 than TripVVT, while TripVVT retains better VFID_R, SSIM, and CLIP-I. The full model improves these latter quantities as well. This supports the viability of the reduced input interface, but differences in backbone initialization and conditioning prevent attributing these gains solely to prior removal.

J.3 Interpreting reward alignment

The MLLM-Reward column is also the signal optimized during post-training and therefore is not an independent measure of its benefit. Table 3 reports positive but modest agreement with human rankings: pairwise agreement is 65.5%, Spearman's ρ is 0.368, and Kendall's τ is 0.310. We accordingly treat this reward as an imperfect training signal rather than a substitute for human evaluation.

The existing post-training ablation in Table 4 provides a complementary view. Relative to the supervised-only model, the full model improves VFID_I, LPIPS, SSIM, CLIP-I, and BG-L1. CLIP-F and BG-DINO change only slightly, while VFID_R increases from 0.1381 to 0.1404, indicating a small deterioration on that metric. Thus the improvement is not uniform across all measures. These are the reported point estimates; they do not establish the statistical significance of small differences or rule out every form of reward exploitation.

The four-method human study in Sec. 4.3 evaluates overall system quality independently of the reward model and gives the full model a 79.4% first-place vote ratio. Because that study compares different systems rather than the same system with and without post-training, it should not be interpreted as isolating the effect of DiffusionNFT. Together, the human evaluation and the independent metrics support the overall try-on performance, while the component ablation provides the more direct evidence about post-training.

J.4 Additional comparisons with general video editors

We additionally evaluate LucyEdit [DecartAI Team, 2025], Ditto [Bai et al., 2025a], and Runway Gen-4 Aleph [Runway, 2025] on the same 100 TripVVT-Bench examples. All methods receive the source video and the editing instruction. For the text-only LucyEdit and Ditto interfaces, an MLLM-generated description of the reference garment is incorporated into the text prompt. Runway receives the reference garment image directly, as does InstructVVT. We use the benchmark metrics defined in Sec. 4.1; Table 9 reports the resulting point estimates.

Table 9: Additional general-editor comparisons on TripVVT-Bench. LucyEdit and Ditto receive garment descriptions, whereas Runway and InstructVVT receive the garment image. MLLM-Reward is the reward used for InstructVVT post-training, rather than an independent evaluation signal.

Method	$\text{VFID}_I \downarrow$	$\text{VFID}_R \downarrow$	SSIM↑	LPIPS↓	CLIP-I↑	BG-L1↓	Reward↑
LucyEdit	31.01	0.7361	0.7081	0.1754	0.8963	0.1528	0.4640
Ditto	41.13	3.6480	0.6223	0.2525	0.8629	0.3075	0.3791
Runway Aleph	22.08	3.4473	0.8303	0.1373	0.9167	0.1778	0.6160
InstructVVT	12.86	0.1404	0.8742	0.0615	0.9528	0.0468	0.7193

InstructVVT obtains better values on the reported metrics in this comparison. The text-only methods operate under a different garment-input protocol, so their results do not isolate architectural differences from the information available at inference. Runway accepts the same input modalities as InstructVVT, but the systems also differ in training and model design. The comparison therefore supports the task-specific system's performance under the stated protocols rather than attributing the entire gap to any single component. The model is designed and evaluated for reference-based garment replacement; these results do not establish general-purpose object editing.

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 state the intended scope as instruction-driven reference-based video virtual try-on with source video, reference garment, and instruction inputs. The claimed MLLM conditioning, garment-reference branch, reward post-training, benchmarks, and ablations are reflected in the method and experiments.

   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: Appendix A discusses limitations related to MLLM and reward quality, extremely long videos, severe occlusion, unusual garment geometry, and instructions requiring large physical changes.

   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 is an empirical method paper and does not present formal theoretical results, theorems, 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: The paper discloses the architecture, conditioning pathways, data construction, training stages, benchmarks, metrics, baseline protocols, ablation variants, reward prompt, and compute settings needed to reproduce the experimental protocol and verify the main claims. Some derived assets, code, and checkpoints are subject to license and company review and may not be publicly released at submission time, but the experimental setup and evaluation protocol are described in the paper and appendix.

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: Open release of data, code, and models must undergo company review and risk assessment. We therefore cannot guarantee open access at submission time.

   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 main paper and appendix specify the training corpus, evaluation benchmarks, baseline protocols, metrics, inference settings, optimizer, learning rate, batch sizes, training stages, step counts, EMA settings, LoRA/query-token settings, reward prompt, and ablation variants. Low-level engineering details such as absolute paths, cache locations, service ports, and dataloader internals are omitted because they do not define the reported experimental protocol.

   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: [No]

Justification: The experiments report benchmark metrics, human preference results, and reward-human agreement, but do not currently include error bars, confidence intervals, or formal 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: [No]

   Justification: The appendix reports H100 GPU counts for the SFT and RL stages. It does not yet provide memory, wall-clock time, storage requirements, per-run compute, or total compute estimates for all experiments.

   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 is designed and reported with attention to consent-sensitive video editing risks, responsible release, and anonymized evaluation. We have reviewed the NeurIPS Code of Ethics and do not identify a required deviation.

   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: Appendix G discusses positive applications in online retail, digital content creation, and accessibility, as well as negative misuse risks such as non-consensual appearance editing, misleading media, and edited videos involving identifiable people. It also discusses consent, disclosure, restricted use, safety filtering, and provenance or watermarking as responsible-deployment considerations.

    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: [No]

    Justification: Any release of data, code, or models must undergo company review and risk assessment, and open release is not guaranteed. As a result, the paper does not yet describe finalized release safeguards.

    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 cites the datasets, models, baselines, tools, and commercial services used in the work, and Appendix H lists their roles and publicly stated license or service-term status. For assets without a standard open-source license, the paper records the official terms status rather than inferring one, and states that any release of derived data, generated results, checkpoints, or code will be reviewed against the applicable terms.

    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: [No]

    Justification: New data, code, or model assets cannot be guaranteed for release because open access requires company review and assessment. Therefore, asset documentation alongside a release is not yet provided.

    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: Appendix D reports the human preference evaluation protocol, including sample counts, compared methods, randomization, ranking criteria, participant-facing instructions, the information shown in each task, the compensation policy, and the fact that no personally identifying information is collected. The evaluation uses preference rankings over anonymized generated videos and does not require screenshots to understand the protocol.

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: [No]

    Justification: The current draft does not describe IRB approval or an equivalent institutional review process for the human preference evaluation. It also does not yet provide a participant-risk disclosure statement.

    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 method explicitly uses MLLMs for conditioning, instruction annotation, and reward scoring. The main paper and appendix describe the MLLM inputs, connector, reward backbone, and fixed reward prompt.

    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.