

BooM-VVT: Boosting Mask-Free Video Virtual Try-On with Image-Level Pseudo Data

Anonymous Author(s)
Submission Id: 4444

SUPPLEMENTARY CONTENTS

Supplementary Contents

A	Related Work	1
A.1	Image Virtual Try-On	1
A.2	Video Virtual Try-On	1
B	Preliminary	2
B.1	Virtual Try-On Paradigms	2
B.1.1	Mask-based and Mask-free Try-on.	2
B.1.2	Keyframe-Driven Video Try-on.	2
B.2	Diffusion Transformers	2
C	Method Details	2
C.1	Keyframe Sampling Algorithm	2
	Implementation note.	3
C.2	Evaluation Metrics	4
C.2.1	VLM-based Evaluation Metrics	4
C.2.2	Discussion on VFID Evaluation	4
C.3	Pseudo-Label Data Construction	4
D	More Results	6
D.1	Additional Quantitative Results	6
D.2	Additional Qualitative Results	6
D.3	Results on Diverse Try-on Tasks	6
D.4	Additional Ablation Results	6
D.4.1	Ablation on OmniView Dataset.	6
D.4.2	Sensitivity to the weighting factor λ	7
	References	13

A Related Work

A.1 Image Virtual Try-On

Image virtual try-on aims to synthesize a realistic image of a target person wearing a reference garment while preserving identity and scene context. Early methods [12, 22, 25] mainly followed a warping-and-fusion pipeline, where the garment was first aligned to the target body via TPS or learned flow and then blended into the person image by a GAN-based generator. Although effective in controlled settings, these methods often suffer from misalignment artifacts and limited robustness under complex poses, severe occlusions, and unconstrained scenes.

Recent advances in diffusion models [14, 19, 21] have substantially improved synthesis realism and garment fidelity in image virtual try-on. Representative methods [4, 5, 15, 26] build on latent diffusion model [21] and further enhance garment preservation through reference-feature injection or simple input concatenation. Beyond standard frontal-view try-on, MV-VTON [23] extends image virtual try-on to multi-view settings by using front- and back-view garment images, and constructs the MVG dataset for this task. However, most existing image try-on methods still rely on dense

preprocessing, such as human parsing and garment masks, which limits robustness and practical usability in real-world scenarios. To alleviate this limitation, more recent studies [10, 11, 28] explore mask-free paradigms, where the model implicitly learns try-on region localization from large-scale pseudo-paired data. As a result, these methods exhibit improved flexibility and generalization. However, their effectiveness has been demonstrated primarily in image synthesis, and extending mask-free paradigms to temporally coherent video generation remains challenging.

A.2 Video Virtual Try-On

Compared with image virtual try-on, video virtual try-on (VVT) additionally requires temporal coherence across frames. Early VVT methods typically extend image-based pipelines to the video setting by introducing dedicated temporal modules. For example, FW-GAN [8] extends image-based try-on to videos through optical-flow-guided warping and fusion, while MV-TON [29] improves temporal consistency with a memory refinement module that propagates information from previously generated frames. Although these methods can produce plausible try-on results, their ability to maintain temporal coherence is often limited by the generative capacity of GAN-based architectures.

With the success of diffusion models in video generation, recent VVT methods have made significant progress. ViViD [9] adapts image diffusion models to video try-on by introducing temporal modeling modules and a dedicated garment encoder. CatV2TON [6] adopts a video DiT architecture to unify image and video try-on within a single model through temporal concatenation of garment and person inputs. MagicTryOn [16] introduces a garment-preservation strategy that decomposes garment cues into semantic, structural, and appearance streams, and further incorporates garment-aware spatiotemporal RoPE to reduce temporal jitter. Despite these advances, most existing VVT methods still rely heavily on scarce paired video try-on data, which limits their generalization to unconstrained real-world scenarios.

To reduce this dependence, DreamVVT [30] shows that a keyframe-driven, stage-wise design can better exploit readily available unpaired videos to improve video-model generalization. KeyTailor [13] introduces a keyframe-driven detail injection strategy to enrich garment dynamics and background integrity. Nevertheless, these methods still depend on explicit masks, which restrict garment fidelity and temporal coherence in challenging real-world scenarios. PEMF-VTO [3] explores mask-free video virtual try-on by introducing point-enhanced guidance for spatial garment transfer and temporal coherence. However, its learning paradigm still relies on large-scale video-level pseudo data. In contrast, our method achieves mask-free VVT under the keyframe-driven paradigm using more accessible image-level pseudo data, substantially reducing the dependence on video-level pseudo data. We further improve

garment consistency in keyframe-driven try-on video generation through garment-sensitive keyframe sampling and explicit alignment between keyframes and target video frames.

B Preliminary

B.1 Virtual Try-On Paradigms

B.1.1 Mask-based and Mask-free Try-on. Virtual try-on methods can be broadly categorized into mask-based and mask-free paradigms. Mask-based methods formulate try-on as a conditional inpainting problem, where an agnostic mask is used to specify the editable clothing region. Given a person image I , a garment image I_g , and a mask M , the try-on result $\hat{I}$ can be written as

$$\hat{I} = G(I, I_g, M), \quad (1)$$

where G denotes the try-on model. This design provides explicit spatial controllability, but its performance heavily depends on mask quality. In complex scenarios, inaccurate masks may destroy original appearance cues and introduce artifacts. By contrast, mask-free methods do not rely on explicit masks and instead require the model to implicitly localize the try-on region from data. The try-on process can be formulated as

$$\hat{I} = G(I, I_g). \quad (2)$$

B.1.2 Keyframe-Driven Video Try-on. Existing video virtual try-on methods are often trained in an end-to-end manner, directly generating a try-on video from garment and video conditions. Given an input video $V = \{I_t\}_{t=1}^T$ and a garment image I_g , the generated try-on video $\hat{V}$ can be written as

$$\hat{V} = G_{\text{vvt}}(V, I_g). \quad (3)$$

Although conceptually simple, this design heavily depends on scarce paired garment–video data, making it difficult to preserve garment fidelity and temporal consistency in unconstrained scenarios. To address this limitation, recent methods adopt a keyframe-driven paradigm, which decomposes VVT into two stages: keyframe try-on and keyframe-guided video generation. Specifically, given sampled keyframes $V_k = \{I_{t_i}\}_{i=1}^K$, the keyframe try-on results are first generated as

$$\hat{V}_k = G_{\text{img}}(V_k, I_g), \quad (4)$$

and the final try-on video is then synthesized as

$$\hat{V} = G_{\text{vid}}(V, \hat{V}_k). \quad (5)$$

This design makes it easier to exploit powerful image try-on models and pretrained video generation backbones, while also enabling training with more readily available unpaired videos. However, existing keyframe-driven methods still mostly rely on explicit masks during try-on generation, which limits their robustness in challenging real-world scenarios.

B.2 Diffusion Transformers

Diffusion Transformers (DiTs) have emerged as powerful backbones for image and video generation due to their strong scalability and ability to model long-range dependencies.

For image generation, recent representative models such as Qwen-Image and FLUX adopt a multimodal diffusion transformer (MM-DiT) architecture. In general, such models consist of a variational autoencoder (VAE), a text encoder, and an MM-DiT backbone.

The input image is first mapped into a latent space by the VAE, while the text prompt is encoded into semantic embeddings by the text encoder. The MM-DiT backbone is composed of stacked Transformer blocks, which enable effective interaction across different modalities through self-attention, thereby supporting high-quality image generation and editing. In our framework, the multi-frame try-on model is built upon Qwen-Image-Edit, whose backbone also follows the MM-DiT architecture.

For video generation, recent mainstream models such as Wan adopt a DiT-based latent video generation architecture. A typical video DiT consists of a VAE, a text encoder, and a DiT backbone that includes patchifying and unpatchifying modules together with multiple Transformer blocks. By operating on spatiotemporal latent tokens, the model can jointly capture temporal continuity and appearance consistency across frames. To better incorporate textual instructions in long-context scenarios, such architectures often combine self-attention and cross-attention to inject conditioning information. In our framework, the video generation model is based on Wan-Animate, which is also built on the video DiT architecture.

Recent DiT-based generative models are often trained with *flow matching*. Let x_1 denote the latent representation of a target sample and $x_0 \sim \mathcal{N}(0, I)$ denote a Gaussian noise sample. Given a sampled time step $t \in [0, 1]$, the interpolated latent is defined as

$$x_t = tx_1 + (1 - t)x_0. \quad (6)$$

The corresponding target velocity field is

$$v_t = \frac{dx_t}{dt} = x_1 - x_0. \quad (7)$$

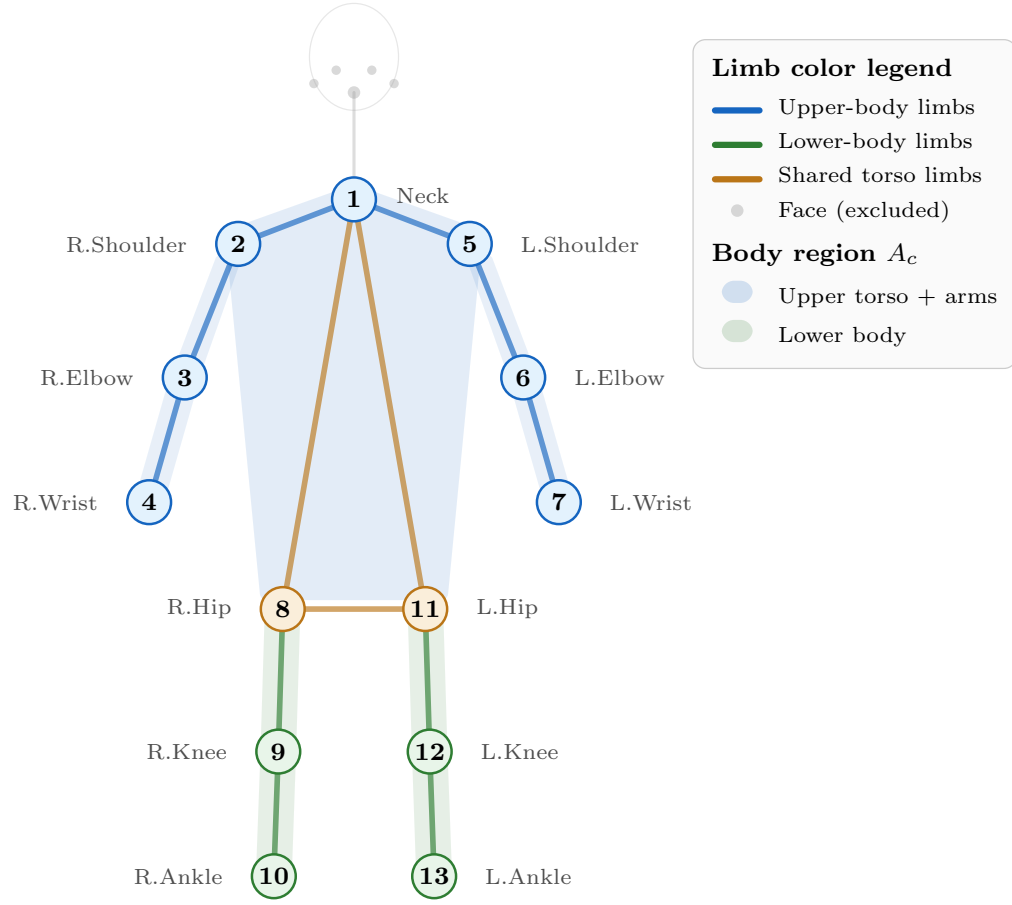
Flow matching trains the network to predict this velocity field under the given conditions, enabling the model to learn a continuous transport from noise to data. In our framework, both the multi-frame try-on model and the video generation model are trained with *flow matching*.

C Method Details

C.1 Keyframe Sampling Algorithm

Fig. 1 shows the DWPose skeleton used in GSKS, together with the definitions of the garment-related limb set P_c and body region A_c for each garment category. Face-related keypoints (indices 0 and 14–17) are excluded from P_c for all garment categories, as they are irrelevant to garment appearance. As described in Sec. 3.2.1 of the main paper, GSKS selects keyframes according to two criteria: *information capacity* and *complementary viewpoint coverage*. Specifically, for upper-body garments, P_c includes the shoulder and arm limbs together with torso connections; for lower-body garments, P_c includes the hip and leg limbs together with torso connections; and for full-body garments, P_c includes all body limbs. In all experiments, we use DWPose [27] for limb detection and SAM [2] for body region estimation. Unless otherwise specified, we set the number of keyframes to $K = 2$, the weighting factor to $\lambda = 0.2$, and the minimum frame interval between selected keyframes to $\Delta_{\min} = 4$ frames.

To facilitate reproducibility, we provide the complete pseudocode of GSKS in Algorithm 1. In the complementary viewpoint selection stage, the viewpoint difference between a candidate frame and the selected keyframes is measured using the average cosine distance



Category c	Garment-related limb set P_c (keypoint pairs)	Body region A_c
■ Upper-body (tops, jackets)	(2,3) (3,4) (5,6) (6,7) (1,2) (1,5) (8,11) (1,8) (1,11)	Upper torso + arms
■ Lower-body (pants, skirts)	(8,9) (9,10) (11,12) (12,13) (8,11) (1,8) (1,11)	Lower body
■ Full-body (dresses)	All of the above (9 + 4 = 13 limb connections)	Full body

Figure 1: DWPose 18-point skeleton and the definitions of the garment-related limb set P_c and body region A_c for each garment category used in GSKS.

of garment-related limb direction vectors. Missing limbs are treated as maximally similar to avoid overestimating viewpoint differences caused by unreliable detections under severe occlusion.

Implementation note. The information-capacity formula in the main paper contains a typo in the weighting coefficients of S_p^c and

S_a^c . The correct formulation, which is used in our implementation and in all experiments, is

$$S_{\text{Capacity}}^c(i) = (1 - \lambda)S_p^c(i) + \lambda S_a^c(i), \quad (8)$$

with $\lambda = 0.2$. This typo does not affect any reported results.

Algorithm 1: Garment-Sensitive Keyframe Sampling (GSKS)

Input: V_{in} : input video (N frames);
 c : target garment category;
 K : number of keyframes;
 $\lambda = 0.2$: weight for region visibility;
 $\Delta_{min} = 4$: minimum frame interval between selected keyframes;
Output: $\mathcal{K}$: selected keyframe set;

- 1 Determine garment-related limb set P_c and body region A_c according to c ;
- 2 */* Phase 1: Per-frame information-capacity scoring */*
- 3 **for** $i = 0$ **to** $N-1$ **do**
- 4 Detect limbs in $V_{in}[i]$ using DWPose;
- 5 **foreach** limb $l = (p_1, p_2) \in P_c$ **do**
- 6 **if** l is detected **then**
- 7 $d_i^l \leftarrow \text{normalize}(p_2 - p_1)$; $\text{valid}_i^l \leftarrow \text{True}$;
- 8 **else**
- 9 $d_i^l \leftarrow 0$; $\text{valid}_i^l \leftarrow \text{False}$;
- 9 Compute limb visibility $S_p^c(i) = \frac{|\hat{P} \cap P_c|}{|\hat{P}|}$, where $\hat{P}$ is the detected limb set in frame i ;
- 10 Estimate garment-related body region A_c using SAM and compute $S_a^c(i) = \frac{\text{area}(A_c)}{\text{area}(V_{in}[i])}$;
- 11 Compute information capacity $S_{\text{Capacity}}^c(i) = (1 - \lambda)S_p^c(i) + \lambda S_a^c(i)$;
- 12 */* Phase 2: Anchor keyframe selection */*
- 13 $i^* \leftarrow \arg \max_i S_{\text{Capacity}}^c(i)$;
- 14 $\mathcal{K} \leftarrow \{V_{in}[i^*]\}$; $\mathcal{I}_{\text{sel}} \leftarrow \{i^*\}$;
- 15 */* Phase 3: Iterative complementary selection */*
- 16 **for** $r = 1$ **to** $K-1$ **do**
- 17 $i_{\text{best}} \leftarrow \text{None}$; $\text{score}_{\text{best}} \leftarrow -\infty$;
- 18 **foreach** candidate frame $i \notin \mathcal{I}_{\text{sel}}$ **do**
- 19 **if** $\exists j \in \mathcal{I}_{\text{sel}} : |i - j| < \Delta_{min}$ **then**
- 20 **continue**;
- 21 **foreach** $j \in \mathcal{I}_{\text{sel}}$ **do**
- 22 $\text{sim}(i, j) \leftarrow \frac{1}{|P_c|} \sum_{l \in P_c} \begin{cases} \cos(d_i^l, d_j^l), & \text{if } \text{valid}_i^l \wedge \text{valid}_j^l; \\ 1.0, & \text{otherwise} \end{cases}$;
- 23 $\text{diff}(i) \leftarrow 1 - \max_{j \in \mathcal{I}_{\text{sel}}} (\text{sim}(i, j))$;
- 24 **if** $(\text{diff}(i), S_{\text{Capacity}}^c(i)) > \text{score}_{\text{best}}$ **then**
- 25 $i_{\text{best}} \leftarrow i$; $\text{score}_{\text{best}} \leftarrow (\text{diff}(i), S_{\text{Capacity}}^c(i))$;
- 26 **if** $i_{\text{best}} \neq \text{None}$ **then**
- 27 $\mathcal{K} \leftarrow \mathcal{K} \cup \{V_{in}[i_{\text{best}}]\}$;
- 28 $\mathcal{I}_{\text{sel}} \leftarrow \mathcal{I}_{\text{sel}} \cup \{i_{\text{best}}\}$;
- 29 **else**
- 30 **break**;
- 31 **return** $\mathcal{K}$;

C.2 Evaluation Metrics

C.2.1 VLM-based Evaluation Metrics. To comprehensively evaluate the quality of generated try-on videos, we introduce three VLM-based metrics: *Appearance & Background Consistency* (ABC),

Garment Consistency (GC), and *Overall Visual Quality* (OVQ). Inspired by recent studies [17, 20] showing strong alignment between large multimodal models and human judgments, we design a structured evaluation pipeline using Gemini 3.1 Pro as the evaluator.

For each test sample, the evaluator receives three inputs: (1) the source model video, which serves as the reference for person identity, motion, and background; (2) the target garment image; and (3) the generated try-on video to be evaluated. The evaluator is instructed to assess the three dimensions independently, each on an integer scale from 0 (worst) to 5 (best). To ensure consistent and reproducible scoring, we provide detailed sub-criteria and scoring anchors for each dimension in the evaluation prompt, as shown in Fig. 2. We set the decoding temperature to 0 and require the output to follow a structured JSON format containing the three integer scores together with brief rationales for each dimension.

C.2.2 Discussion on VFID Evaluation. Besides the FVD metric used in the main paper, prior VVT works have also widely reported the Video Fréchet Inception Distance (VFID) [8] for quantitative comparison. Although we do not use VFID as a primary metric in the main paper, we include it here for completeness and comparability with prior work.

During our study, we identified an implementation issue in a commonly used open-source VFID toolkit¹ adopted by several recent methods [6, 16]. Specifically, regardless of the video length, the current implementation evaluates only the first 10 frames of each video and ignores all subsequent frames. As a result, the metric cannot fully assess quality over the entire video sequence, especially for videos with large motion variation or rapidly changing content. We provide a detailed analysis and a corrected implementation in the supplementary document `vfid_implementation_analysis.pdf`. Based on the corrected implementation, we report VFID scores for all compared methods in Sec. D.

C.3 Pseudo-Label Data Construction

As described in the main paper, we use two types of pseudo data in the multi-stage training strategy: image-level pseudo data for Stage 2 and video-level pseudo data for Stage 3. Their construction pipelines are detailed below.

Image-level pseudo data. For each person image, we construct a corresponding pseudo image through the following steps. (1) We first use Qwen3-VL-32B [1] to analyze the garment currently worn by the person and generate a garment editing instruction (e.g., “change the striped t-shirt to a plain white hoodie”). (2) We then apply Qwen-Image-Edit-2511 [24] to modify the garment region according to the instruction, producing a pseudo image. (3) Finally, we use Qwen3-VL-32B to verify whether the generated pseudo image is consistent with the editing instruction. Samples that fail verification are discarded. To accelerate inference, we adopt Qwen-Image-Edit-Lightning [18], a LoRA-based acceleration module for Qwen-Image-Edit. In total, we construct 30K pseudo images from person images covering diverse garment types, body poses, and real-world backgrounds.

Video-level pseudo data. Since existing open-source video editing models cannot produce sufficiently high-quality try-on videos, we

¹<https://github.com/ZhengJun-AI/vfid-metrics>, `fid_metrics/dataset.py`, lines 106–139.

System Prompt

Task Definition

You are a "Virtual Try-on Video Quality Evaluator". Your task is to objectively and reproducibly score a given generated try-on video. You will receive three inputs:

1) **Source Model Video**: the original model video (reference for person identity, motion, and background).

2) **Garment Image**: the target garment image (reference for the desired garment).

3) **Generated Try-on Video**: the try-on video produced by a method (the object to be evaluated).

Please score each of the following **3 dimensions** independently. Each criterion is scored on a scale of **0 to 5**, with 0 indicating the worst and 5 the best. Scores must be integers: 0/1/2/3/4/5.

Dimension 1: Appearance & Background Consistency (ABC, 0–5)

Sub-criteria

- **Model identity stability**: Whether face, hairstyle, skin tone, and body shape remain consistent; whether identity swap, drift, or deformation occurs.
- **Motion coherence**: Whether motions are natural and continuous; whether the overall motion style is consistent with the source video.
- **Non-try-on region consistency**: Whether the appearance of regions outside the garment try-on area (e.g., head, hands, legs) remains stable without abnormal re-rendering.
- **Background consistency**: Whether scene elements, camera viewpoint, and background details remain stable; whether background distortion, jumping, re-rendering, or noticeable jittering occurs.

Scoring Anchors

5: Identity stable; non-try-on regions almost unchanged; motions natural and coherent; background stable with no distortion or jumping.

4: Overall consistency very high; minor jittering or detail fluctuations that do not affect viewing.

3: Generally consistent; visible issues exist (mild drift, occasional deformation, slight background instability), but still acceptable.

2: Poor consistency; multiple noticeable identity or background issues, motion incoherence, or artifacts that affect viewing.

1: Severely inconsistent; frequent face swaps, deformations, background collapse, or clearly abnormal motions.

0: Almost unrecognizable as the same person or scene; video essentially unusable.

Dimension 2: Garment Consistency (GC, 0–5)

Sub-criteria

- **Garment presence**: Whether the target garment is correctly worn in the right category and position (not pasted onto the background or misplaced on the body), without frequent disappearance.
- **Style & structure consistency**: Whether the silhouette, neckline, sleeve type, garment length, closure style, pockets/buttons/zippers and other key structural elements match the reference.
- **Pattern/texture/color consistency**: Whether patterns, material textures, logo/print placement, and colors are accurate and temporally stable across frames (no flickering, drifting, or "style shifting").
- **Motion-following plausibility**: Whether garment deformations and wrinkles in response to body movements are reasonable, without noticeable stretching, tearing, or clipping.

Scoring Anchors

5: Target garment clearly worn; structural details highly consistent; patterns/textures/colors accurate and stable across frames.

4: Garment correctly worn and closely resembles reference; minor detail deviations or slight jittering/blurring that do not affect recognition.

3: Recognizable as the target garment but with noticeable differences (structure/pattern position/color shifting or flickering); overall acceptable.

2: Poor consistency; key structural errors or frequent pattern/texture drifting, deformation, or severe blurring; easily misidentified.

1: Almost incorrectly dressed; garment category/structure clearly wrong, or frequently disappears/breaks.

0: Target garment not generated or unrecognizable as a garment try-on result.

Dimension 3: Overall Visual Quality (OVQ, 0–5)

Sub-criteria

- **Realism & fusion naturalness**: Whether the garment blends naturally with the body and background; whether edge transitions are clean; whether a "pasted-on" appearance exists.
- **Lighting/shadow/occlusion plausibility**: Whether lighting is consistent; whether shadows and highlights are reasonable; whether occlusion relationships are correct.
- **Temporal consistency & smoothness**: Whether inter-frame flickering, jittering, or jumping occurs; whether motions are coherent; whether the garment stably follows body movements.
- **Artifacts & defects**: Severity and frequency of blurring, ghosting, mosaic patterns, tearing, stretching, local collapse, and other visual anomalies.

Scoring Anchors

5: Very natural and realistic appearance; excellent fusion; temporally stable; no noticeable artifacts.

4: Good visual quality; minor imperfections that do not affect the overall experience.

3: Moderately usable; noticeable but tolerable issues.

2: Poor visual quality; frequent artifacts, flickering, tearing, or blurring.

1: Severe defects; large-area artifacts or frequent collapse; nearly unusable.

0: Complete video failure (persistent collapse, unwatchable, or unrecognizable content).

Output Format (output JSON only, no extra text):

```
{
  "score_background_model_consistency": 0-5,
  "score_garment_consistency": 0-5,
  "score_overall_quality": 0-5,
  "rationale": { ... 1-3 sentences per dimension ... }
}
```

Figure 2: Evaluation prompt used for VLM-based video virtual try-on assessment. The evaluator is instructed to score each generated try-on video along three dimensions: Appearance & Background Consistency (ABC), Garment Consistency (GC), and Overall Visual Quality (OVQ), each on an integer scale from 0 to 5 with detailed scoring anchors.

Table 1: Per-sample construction cost of pseudo data. All timings are measured on a single NVIDIA A800 GPU.

Data Type	Resolution	Frames	Image Editing / Try-on	Video Generation	Total
Image	832×624	1	4 s	–	4 s
Video	832×624	65	12 s	262 s	274 s

Table 2: VFID evaluation using the corrected implementation. VFID_I: I3D backbone; VFID_R: ResNeXt backbone. ↓: lower is better. Bold: best; Underline: second best.

Method	ViViD-S		WildVVT	
	VFID _I ↓	VFID _R ↓	VFID _I ↓	VFID _R ↓
ViViD [9]	9.36	<u>0.20</u>	16.86	0.33
CatV ² TON [6]	9.84	0.26	28.27	6.05
MagicTryOn [16]	9.28	0.21	14.84	0.51
BooM-VVT (Stage 1)	<u>9.24</u>	0.24	<u>14.59</u>	0.49
BooM-VVT	9.16	0.18	13.96	<u>0.47</u>

use our Stage 1 model, BooM-VVT (Stage 1), to synthesize pseudo try-on videos. The construction pipeline is as follows. (1) Given a source video and a randomly selected target garment image, we apply GSKS to select keyframes from the source video. (2) We generate the corresponding keyframe try-on images using the multi-frame try-on model. (3) The source video, together with the keyframe try-on images and garment masks, is then fed into BooM-VVT (Stage 1) to generate a try-on video. (4) Finally, we use Qwen3-VL-32B to verify the generated video and ensure its quality. Samples with noticeable artifacts or garment inconsistencies are discarded. In total, we construct 2.5K video-level pseudo samples from unpaired human-centric videos.

Cost analysis. Table 1 compares the per-sample construction cost of the two data types. For image-level pseudo data, each sample requires only ~4 s for image editing. For video-level pseudo data, each sample requires ~12 s for keyframe try-on image synthesis and ~262 s for video generation, totaling ~274 s. Overall, constructing 30K image-level pseudo samples costs 33.3 GPU hours, while constructing 2.5K video-level pseudo samples costs 190.3 GPU hours. On a per-sample basis, image-level pseudo data is approximately 69× cheaper. This favorable cost ratio motivates our strategy of leveraging large-scale image-level pseudo data to expand data coverage at low cost, substantially reducing the reliance on expensive video-level pseudo data.

D More Results

D.1 Additional Quantitative Results

Using the corrected VFID implementation described in Sec. C.2.2, we report the VFID results on the ViViD and WildVVT datasets in Table 2. BooM-VVT achieves the best results in three of the four settings, and ranks second in the remaining one. In particular, on the more challenging WildVVT benchmark, BooM-VVT achieves the lowest VFID_I by a clear margin, indicating stronger generalization in unconstrained real-world scenarios.

D.2 Additional Qualitative Results

We provide additional qualitative comparisons on the WildVVT benchmark to further evaluate BooM-VVT under challenging real-world scenarios, including large motion, severe occlusion, and object interaction.

Fig. 3 presents qualitative results under large motion. Compared with existing methods, BooM-VVT produces more stable garment appearance and better temporal consistency. In the left example of Fig. 3, the full mask-free model, BooM-VVT, also achieves better garment consistency than its mask-dependent variant, BooM-VVT (Stage 1), indicating that inaccurate masks caused by fast motion can degrade try-on region localization.

Fig. 4 shows results under severe occlusion. In such cases, heavy self-occlusion makes garment transfer particularly challenging for existing methods, often leading to obvious artifacts or incorrect garment structures. By contrast, BooM-VVT maintains plausible garment appearance even when large body regions are occluded.

Fig. 5 illustrates results in scenarios involving complex interactions between the person and surrounding objects. BooM-VVT better preserves the original interaction relationships, whereas existing methods often introduce noticeable artifacts in the interaction regions and may even distort or erase the interacting objects.

D.3 Results on Diverse Try-on Tasks

Enabled by the proposed OmniView dataset, BooM-VVT supports diverse virtual try-on tasks, including layered try-on, cross-category try-on, and multi-view try-on. Fig. 6, Fig. 7, and Fig. 8 show representative results for these three settings, demonstrating that BooM-VVT generalizes well across diverse try-on scenarios.

Specifically, for the **layered try-on** task (Fig. 6), following DiOr [7], we perform multiple keyframe try-on operations to sequentially dress garments layer by layer, from innerwear to outerwear. For the **cross-category try-on** task (Fig. 7), we provide additional garment information in the text prompt to guide the model toward plausible cross-category try-on results. For the **multi-view try-on** task (Fig. 8), we concatenate garment images from different view-points and feed them into the multi-frame try-on model in a single forward pass. As shown in Fig. 8, BooM-VVT already produces reasonable results with only a front-view garment image. When multi-view garment images are additionally provided, the model generates more accurate results, especially for back-view frames.

D.4 Additional Ablation Results

D.4.1 Ablation on OmniView Dataset. To validate the effectiveness of the proposed OmniView dataset, we compare the full model with a variant whose multi-frame try-on model is not finetuned on OmniView (w/o OmniView). As shown in Fig. 9, training with OmniView brings two key improvements.

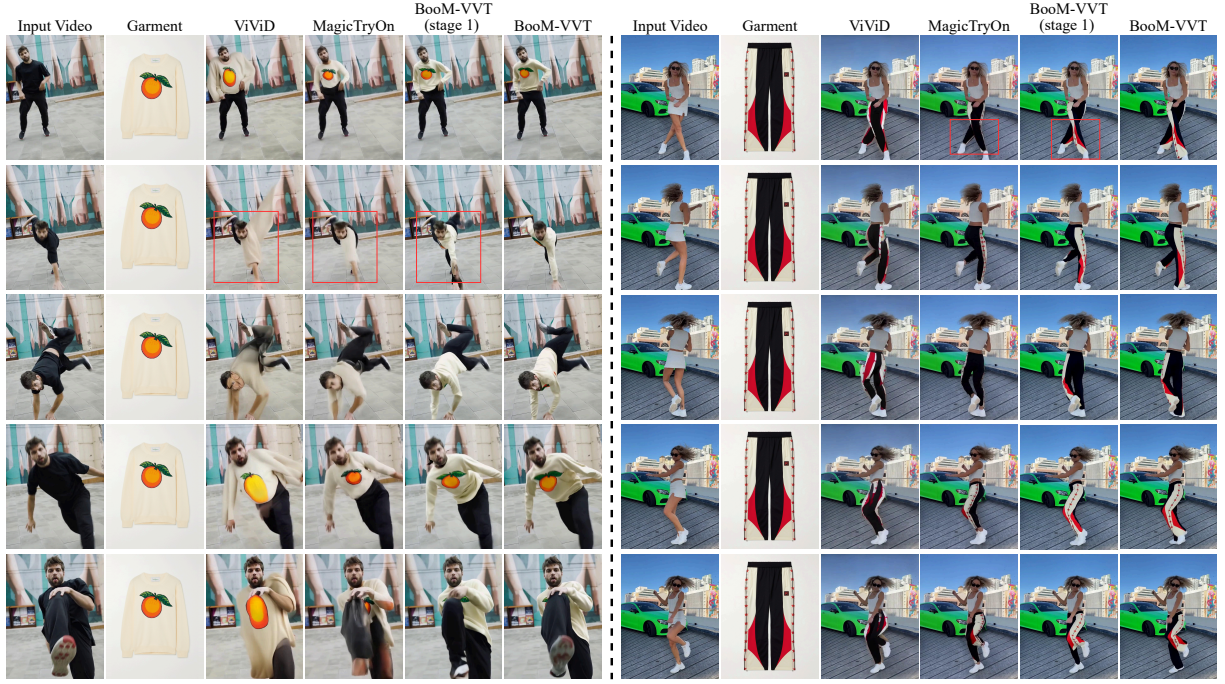


Figure 3: Qualitative results under large motion. Red boxes highlight artifacts produced by existing methods.

First, OmniView enhances cross-view consistency in keyframe try-on results. As illustrated in the left example of Fig. 9, the *w/o OmniView* variant generates inconsistent garment details across viewpoints, such as a belt that appears in one view but disappears in another. In contrast, the full model produces consistent try-on results across viewpoints.

Second, OmniView significantly improves the model’s capability for multi-view garment inputs. As shown in the right example of Fig. 9, even when both front-view and back-view garment references are provided, the *w/o OmniView* variant still produces incorrect garment textures when the person turns around. In contrast, the full model synthesizes sharper textures and more accurate garment structures that are consistent with the reference, highlighting the importance of high-quality and diverse multi-view try-on data.

D.4.2 Sensitivity to the weighting factor λ . We further study the effect of the weighting factor λ in GSKS, which balances limb visibility and region visibility in the information-capacity score. Following the keyframe-driven setting, we evaluate BooM-VVT (Stage 1) on the WildVVT benchmark by reconstructing videos conditioned on the selected keyframes. The results are shown in Table 3.

Overall, the performance remains stable across a wide range of λ values. In most cases, the model achieves very similar SSIM and LPIPS scores, indicating that GSKS is not overly sensitive to this hyperparameter. In particular, when λ is set between 0.2 and 0.6, the results are consistently strong, with SSIM staying at around 0.871 and LPIPS at around 0.062. When λ becomes too large, performance starts to degrade slightly, suggesting that overemphasizing region

Table 3: Sensitivity analysis of the weighting factor λ in GSKS on the WildVVT benchmark.

λ	SSIM $\uparrow$	LPIPS $\downarrow$
0.1	0.865	0.067
0.2	0.871	0.062
0.3	0.871	0.062
0.4	0.871	0.062
0.5	0.870	0.062
0.6	0.871	0.062
0.7	0.869	0.064
0.8	0.867	0.069
0.9	0.862	0.072

visibility may weaken the contribution of limb-based viewpoint cues. Based on this observation, we set $\lambda = 0.2$ in all experiments.

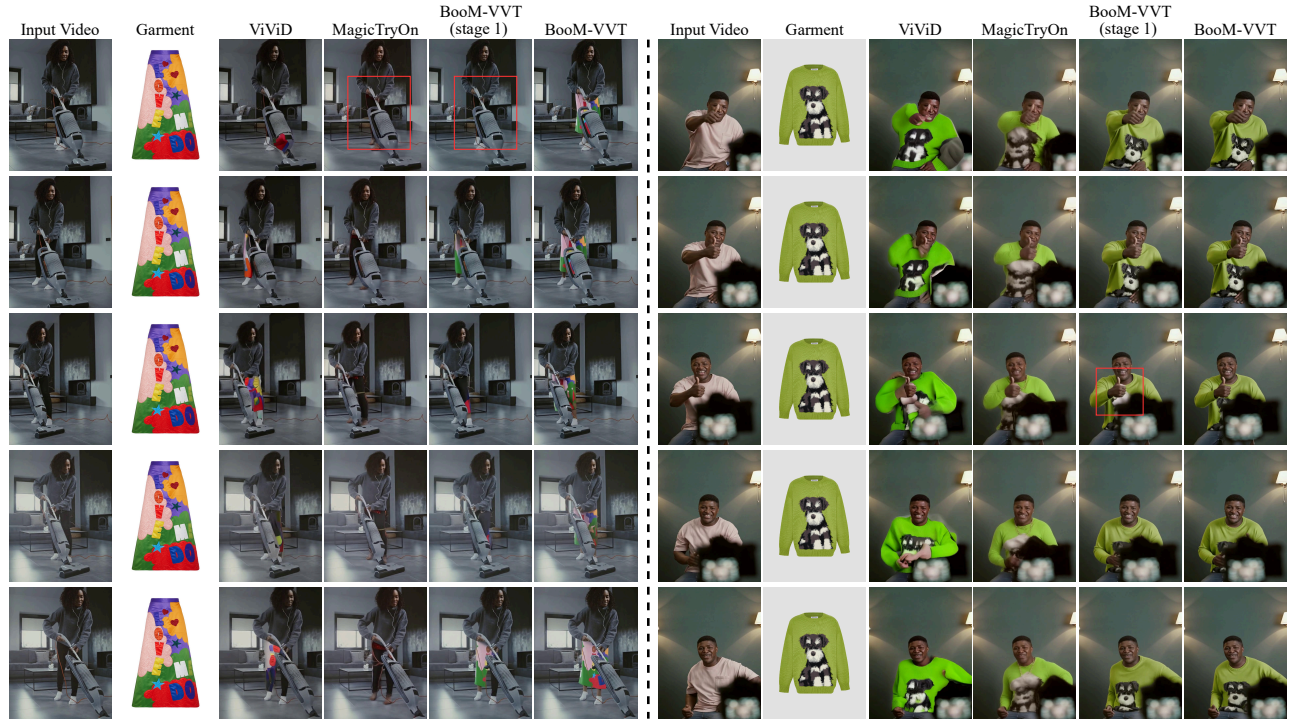


Figure 4: Qualitative results under severe occlusion. Red boxes highlight artifacts produced by existing methods.

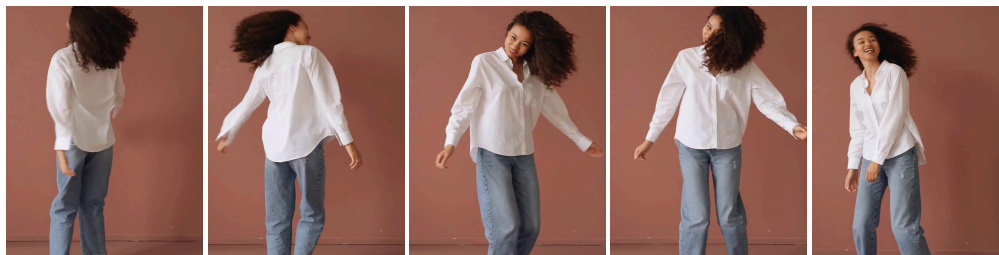


Figure 5: Qualitative results under object interaction. Red boxes highlight artifacts produced by existing methods.

Garment



Input
Video



BooM-
VVT



Garment



Input
Video



BooM-
VVT



Figure 6: Results of BooM-VVT on the layered try-on task.

Garment

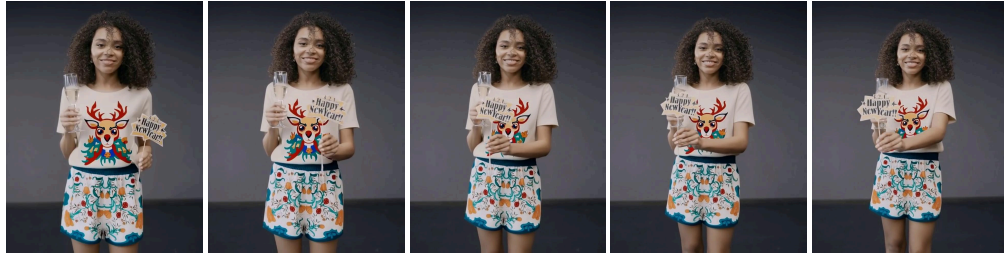


A model wears a beige crew-neck short-sleeve t-shirt with a colorful cartoon reindeer print and matching floral-patterned casual shorts.

Input Video



BooM-VVT

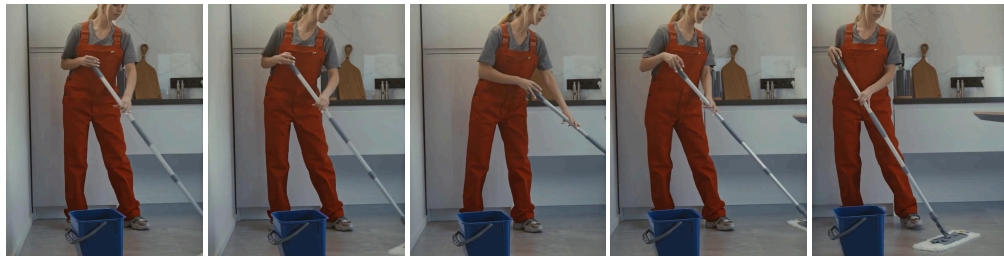


Garment



A model wears a white oversized crew-neck short-sleeve t-shirt with a panda print and a royal blue high-waisted A-line midi skirt.

Input Video



BooM-VVT



Figure 7: Results of BooM-VVT on the cross-category try-on task.



Figure 8: Results of BooM-VVT on the multi-view try-on task. The third and fourth rows show results with single-view (front only) and multi-view (front + back) garment inputs, respectively.

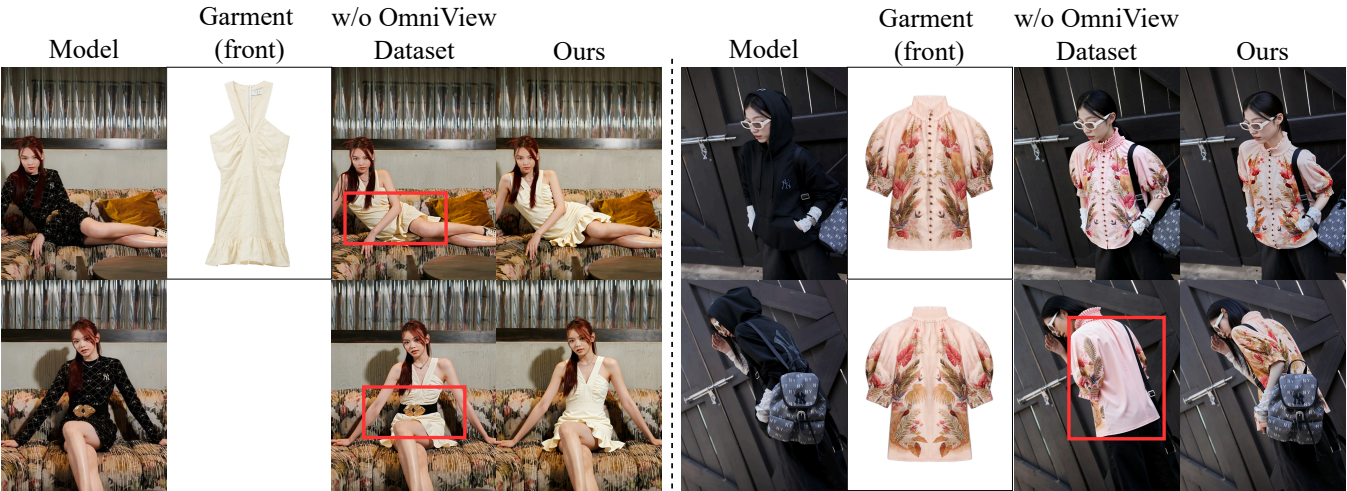


Figure 9: Qualitative ablation on the OmniView dataset. Left: single-view garment input. The w/o OmniView variant produces cross-view inconsistencies (e.g., the belt highlighted by the red box). Right: multi-view garment input (front & back). The w/o OmniView variant generates incorrect textures in back-view frames, while the full model preserves accurate garment details. Please zoom in for details.

References

- [1] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631* (2025).
- [2] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. 2025. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719* (2025).
- [3] Tianyu Chang, Xiaohao Chen, Zhichao Wei, Xuanpu Zhang, Qingguo Chen, Weihua Luo, Peipei Song, and Xun Yang. 2025. PEMF-VTO: Point-Enhanced Video Virtual Try-on via Mask-free Paradigm. *IEEE Transactions on Consumer Electronics* (2025).
- [4] Yisol Choi, Sangkyung Kwak, Kyungmin Lee, Hyungwon Choi, and Jinwoo Shin. 2024. Improving diffusion models for authentic virtual try-on in the wild. In *European Conference on Computer Vision*. Springer, 206–235.
- [5] Zheng Chong, Xiao Dong, Haoxiang Li, Shiyue Zhang, Wenqing Zhang, Xujie Zhang, Hanqing Zhao, Dongmei Jiang, and Xiaodan Liang. 2024. Catvton: Concatenation is all you need for virtual try-on with diffusion models. *arXiv preprint arXiv:2407.15886* (2024).
- [6] Zheng Chong, Wenqing Zhang, Shiyue Zhang, Jun Zheng, Xiao Dong, Haoxiang Li, Yiling Wu, Dongmei Jiang, and Xiaodan Liang. 2025. Catv2ton: Taming diffusion transformers for vision-based virtual try-on with temporal concatenation. *arXiv preprint arXiv:2501.11325* (2025).
- [7] Aiyu Cui, Daniel McKee, and Svetlana Lazebnik. 2021. Dressing in order: Recurrent person image generation for pose transfer, virtual try-on and outfit editing. In *Proceedings of the IEEE/CVF international conference on computer vision*. 14638–14647.
- [8] Haoye Dong, Xiaodan Liang, Xiaohui Shen, Bowen Wu, Bing-Cheng Chen, and Jian Yin. 2019. Fw-gan: Flow-navigated warping gan for video virtual try-on. In *Proceedings of the IEEE/CVF international conference on computer vision*. 1161–1170.
- [9] Zixun Fang, Wei Zhai, Aimin Su, Hongliang Song, Kai Zhu, Mao Wang, Yu Chen, Zhiheng Liu, Yang Cao, and Zheng-Jun Zha. 2024. Vivid: Video virtual try-on using diffusion models. *arXiv preprint arXiv:2405.11794* (2024).
- [10] Yutong Feng, Linlin Zhang, Hengyuan Cao, Yiming Chen, Xiaodan Feng, Jian Cao, Yuxiong Wu, and Bin Wang. 2025. Omnistry: Virtual try-on anything without masks. *arXiv preprint arXiv:2508.13632* (2025).
- [11] Hailong Guo, Bohan Zeng, Yiren Song, Wentao Zhang, Jiaming Liu, and Chuang Zhang. 2025. Any2anytryon: Leveraging adaptive position embeddings for versatile virtual clothing tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 19085–19096.
- [12] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S Davis. 2018. Viton: An image-based virtual try-on network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7543–7552.
- [13] Qingdong He, Xueqin Chen, Yanjie Pan, Peng Tang, Pengcheng Xu, Zhenye Gan, Chengjie Wang, Xiaobin Hu, Jiangning Zhang, and Yabiao Wang. 2025. The devil is in the details: Enhancing Video Virtual Try-On via Keyframe-Driven Details Injection. *arXiv preprint arXiv:2512.20340* (2025).
- [14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.
- [15] Boyuan Jiang, Xiaobin Hu, Donghao Luo, Qingdong He, Chengming Xu, Jinlong Peng, Jiangning Zhang, Chengjie Wang, Yunsheng Wu, and Yanwei Fu. 2024. Fitdit: Advancing the authentic garment details for high-fidelity virtual try-on. *arXiv preprint arXiv:2411.10499* (2024).
- [16] Guangyuan Li, Siming Zheng, Hao Zhang, Jinwei Chen, Junsheng Luan, Binkai Ou, Lei Zhao, Bo Li, and Peng-Tao Jiang. 2025. MagicTryOn: Harnessing Diffusion Transformer for Garment-Preserving Video Virtual Try-on. *arXiv preprint arXiv:2505.21325* (2025).
- [17] Ente Lin, Xujie Zhang, Fuwei Zhao, Yuxuan Luo, Xin Dong, Long Zeng, and Xiaodan Liang. 2025. Dreamfit: Garment-centric human generation via a lightweight anything-dressing encoder. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 5218–5226.
- [18] ModelTC. 2025. *Qwen-Image-Lightning: Speed up Qwen-Image model with distillation*. Retrieved April 2, 2026 from <https://github.com/ModelTC/Qwen-Image-Lightning> GitHub repository. Includes Qwen-Image-Edit-Lightning models, weights, and workflows..
- [19] Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In *International conference on machine learning*. PMLR, 8162–8171.
- [20] Yang Peng, Yuxin Cui, Haomiao Tang, Zekun Qi, Runpei Dong, Jing Bai, Chunrui Han, Zheng Ge, Xiangyu Zhang, and Shu-Tao Xia. 2024. Dreambench++: A human-aligned benchmark for personalized image generation. *arXiv preprint arXiv:2406.16855* (2024).
- [21] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.
- [22] Bochao Wang, Huabin Zheng, Xiaodan Liang, Yimin Chen, Liang Lin, and Meng Yang. 2018. Toward characteristic-preserving image-based virtual try-on network. In *Proceedings of the European conference on computer vision (ECCV)*. 589–604.
- [23] Haoyu Wang, Zhilu Zhang, Donglin Di, Shiliang Zhang, and Wangmeng Zuo. 2025. Mv-vton: Multi-view virtual try-on with diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 7682–7690.
- [24] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Shengming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. 2025. Qwen-image technical report. *arXiv preprint arXiv:2508.02324* (2025).
- [25] Zhenyu Xie, Zaiyu Huang, Xin Dong, Fuwei Zhao, Haoye Dong, Xijin Zhang, Feida Zhu, and Xiaodan Liang. 2023. Gp-vton: Towards general purpose virtual try-on via collaborative local-flow global-parsing learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 23550–23559.
- [26] Yuhao Xu, Tao Gu, Weifeng Chen, and Arlene Chen. 2025. Ootdiffusion: Outfitting fusion based latent diffusion for controllable virtual try-on. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 8996–9004.
- [27] Zhenqiang Yang, Ailing Zeng, Chun Yuan, and Yu Li. 2023. Effective whole-body pose estimation with two-stages distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4210–4220.
- [28] Wei Zhang, Yeying Jin, Xin Li, Yan Zhang, Xiaofeng Cong, Cong Wang, Fengcai Qiao, and Zhichao Lian. 2026. UniFit: Towards Universal Virtual Try-on with MLLM-Guided Semantic Alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 40. 12816–12824.
- [29] Xiaoqing Zhong, Zhonghua Wu, Taizhe Tan, Guosheng Lin, and Qingyao Wu. 2021. Mv-ton: Memory-based video virtual try-on network. In *Proceedings of the 29th ACM International Conference on Multimedia*. 908–916.
- [30] Tongchun Zuo, Zaiyu Huang, Shuliang Ning, Ente Lin, Chao Liang, Zerong Zheng, Jianwen Jiang, Yuan Zhang, Mingyuan Gao, and Xin Dong. 2025. Dreamvvt: Mastering realistic video virtual try-on in the wild via a stage-wise diffusion transformer framework. *arXiv preprint arXiv:2508.02807* (2025).