Vanast: Virtual Try-On with Human Image Animation via Synthetic Triplet Supervision

Supplementary Material

A. Additional Experiments: VVT

A.1. Baselines

Configurations. We design an alternative two-stage pipeline for comparison. In the first stage, the human image $\mathbf{I^G}$ is directly animated using pose guidance $\mathbf{K}$, producing a human animation video without any visual modifications such as upper clothing or dresses. In the second stage, a video virtual try-on model is applied to the animated sequence, yielding a virtual try-on result that preserves human-specific motion patterns and identity. Note that we derive the auxiliary inputs—pose videos, mask videos, and masked videos—from the animation video generated in the first stage and, together with the target garment image, use them as inputs to the corresponding video virtual try-on models to obtain the final animated try-on results.

Details. For video virtual try-on, we adopt (1) MagicTryOn [9], a video virtual try-on model based on large-scale video diffusion Transformers, and (2) SwiftTry [10], a video virtual try-on method that formulates garment transfer as conditional video inpainting using temporally-extended diffusion. For MagicTryOn, we strictly follow the official pipeline and directly generate all auxiliary inputs using code released by the authors, including text captions, pose videos, mask videos and masked videos. For SwiftTry, we prepare the pose videos according to the authors' instructions. However, since no official mask-generation method is provided, we again use mask-generation implementation provided by MagicTryOn. For animation, we employ StableAnimator [14], which consistently demonstrated the highest performance in our experiments. All motion generation across baselines adheres to identical pose sequences to enable controlled comparison.

Metrics and Evaluation Datasets. We use the same metrics for additional comparisons as those introduced in our main paper: L_1, PSNR, SSIM, LPIPS, FID [7], and VFID [5]. We evaluate our method using the same set of benchmarks as in the main paper, employing both the Internet dataset and the ViViD [5] dataset.

A.2. Results

As shown in Tab. 4, our method achieves the best performance across all metrics compared to all baselines. In addition, we provide qualitative comparisons with the baselines in Fig. 19. Our results demonstrate the highest quality in pose accuracy, garment transfer, and identity preservation across all datasets.

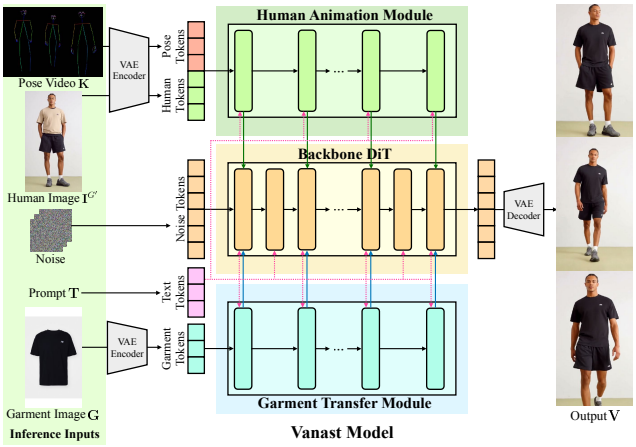

Figure 11. **Inference Pipeline.** We additionally present only the inference pipeline in the main paper overview Fig. 2.

B. Implementation Details

B.1. Model Architecture

We present the model architecture that describes in detail the integration of the backbone DiT with HAM and GTM in Fig. 12. For the HAM and GTM insertion strategy, the modules are attached at the even-indexed backbone blocks following Eq. (2) of our main paper, i.e., $l = 2k$ for $0 \leq k \leq 14$. For the remaining backbone-only blocks, the hidden representation is updated only by the original T2V backbone block as $h_{l+1} = B_l^{\mathrm{T2V}}(h_l)$ if $l \neq 2k$, following Eq. (2) of our main paper. When combining the block outputs, we use fixed weights $\alpha = \beta = 0.5$. Each HAM and GTM module contains approximately 0.65B parameters.

For the projection layer of backbone DiT, we use a `Conv3d` layer with input channel 16 and output channel 1536, with kernel size $(1, 2, 2)$ and stride $(1, 2, 2)$, i.e., `Conv3d(16, 1536, kernel=(1,2,2), stride=(1,2,2))`. The projection layers used in HAM and GTM take 96 input channels, while the output channels, kernel size, and stride are identical to those of the backbone DiT projection layer, following VACE [8]. Patch embedding is implemented using 3D convolutional embedding, and positional encoding is given by 3D RoPE [13].

For token construction, the GTM input tokens are ordered as `torch.cat([Garment token, Zero`

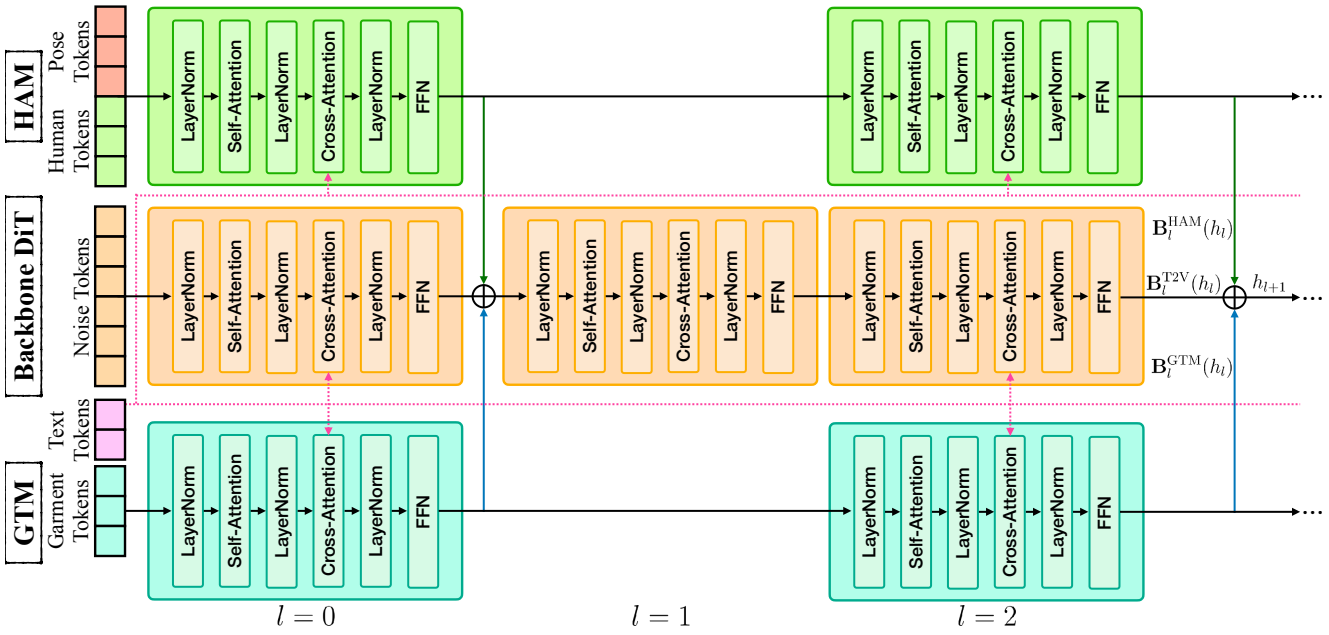

Figure 12. **Model Architecture.** We extend the overview in Fig. 2 of the main paper to present the detailed structure of HAM, GTM, and the backbone DiT.

`placeholder], dim=time)`, while the HAM input tokens are ordered as `torch.cat([Human token, Pose token], dim=time)`.

B.2. Training Details

Training strategy. We train our model based on the Wan2.1 T2V 1.3B backbone weights [15]. Both the Human Animation Module (HAM) and the Garment Transfer Module (GTM) share the same model architecture. We initialize HAM and GTM using the weights from the 0th, 2nd, 4th, 6th, and subsequent even-indexed modules of the Wan2.1 T2V backbone. The projection layer is initialized with weights from VACE [8]. The projection layers of HAM and GTM are trained independently. During training, the backbone DiT weights are frozen, while the HAM and GTM weights remain learnable and are jointly optimized. The VAE encoder and decoder follow the VAE architecture of Wan2.1 and remain frozen during both training and inference. The training is conducted for a total of 10,000 steps.

Hyperparameters. The effective training batch size is set to 32 by applying gradient accumulation. During training, each iteration processes 81 video frames. The learning rate is set to $2e-05$, and the input resolution is fixed to a width of 432 and a height of 768. The maximum number of garment images used per training instance is six. We train our model using eight NVIDIA RTX Pro 6000 96GB GPUs.

Dataset configurations. We train our model, Vanast, on 9135 videos, each ranging from approximately three to ten seconds in duration. The dataset distribution across garment categories is as follows: 4273 upper-body clothing videos, 1335 lower-body clothing videos, 2222 paired upper-body and lower-body videos, 155 hat videos, and 1150 dress videos.

B.3. Loss Functions

Following Wan2.1 model [15], we adopt a flow matching objective to train the continuous-time generative process used in our model. More specifically, let $\mathbf{x}_1$ denote the latent from synthetic triplet datasets, and let $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ be a Gaussian noise sample. A time variable $t \in (0, 1)$ is drawn from a logit-normal distribution. An intermediate latent state by linearly interpolating between $\mathbf{x}_0$ and $\mathbf{x}_1$ is as follows:

$$\mathbf{x}(t) = (1-t)\,\mathbf{x}_0 + t\,\mathbf{x}_1. \tag{4}$$

The ground-truth velocity is then obtained by differentiating $\mathbf{x}(t)$ with respect to t,

$$\mathbf{v}(t) = \frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{x}_1 - \mathbf{x}_0. \tag{5}$$

The model $u(\mathbf{x}(t), \mathbf{c}, t; \theta)$ is trained to predict this velocity, where θ denotes all learnable parameters and $\mathbf{c}$ represents a text-conditioning embedding, umT5 [2]. We optimize a mean squared error objective:

$$\mathcal{L} = \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_1, t, \mathbf{c}} \left[\left\| u(\mathbf{x}(t), \mathbf{c}, t; \theta) - \mathbf{v}(t) \right\|^2 \right]. \tag{6}$$

Figure 13. **Additional Results of Upper-Garment Transfer.** We present virtual upper-garment try-on with human image animation results.

Table 4. **Quantitative Comparison with the Combination of Animation and Video Virtual Try-On Models. Bold text** indicates the best score within each column. Our results demonstrate superior performance across all metrics and all datasets.

		Internet Dataset							ViViD Dataset						
Animation	Video Virtual Try-On	$L_1 \downarrow$	PSNR↑	SSIM↑	LPIPS↓	FID↓	$VFID_{I3D} \downarrow$	$VFID_{ResNeXt} \downarrow$	$L_1 \downarrow$	PSNR↑	SSIM↑	LPIPS↓	FID↓	$VFID_{I3D} \downarrow$	$VFID_{ResNeXt} \downarrow$
StableAnimator	Magic Try-On	0.1032	15.28	0.6863	0.3068	121.50	24.04	0.64	0.2044	10.86	0.5632	0.5287	145.77	40.82	3.21
	SwiftTry	0.1103	14.97	0.7002	0.3238	147.34	27.37	0.93	0.2207	10.51	0.5842	0.5459	157.96	41.16	3.51
Ours		**0.0719**	**17.95**	**0.7550**	**0.2370**	**91.05**	**22.52**	**0.39**	**0.1077**	**14.67**	**0.6686**	**0.3649**	**105.89**	**35.72**	**1.30**

Here, $\mathbf{x}_0$, $\mathbf{x}_1$, and t jointly specify the point along the straight path between noise and data latents, and the model learns to approximate the continuous-time flow that transports $\mathbf{x}_0$ to $\mathbf{x}_1$. This formulation is equivalent to maximum-likelihood training under rectified flows [4], while allowing stable optimization through ordinary differential equation solvers.

B.4. Inference Details

To construct the textual prompts required for both training and inference, we employ a vision–language model (VLM) [1]. We uniformly sample eight frames from each input video and provide them to the VLM with the following instruction: *"Describe this video in detail without repetition, excluding any descriptions related to clothing such as tops, lower-body garments, or outerwear."* The resulting descriptions are subsequently refined using a large language model (LLM) [6], allowing us to obtain clean and consistent text prompts that capture motion and appearance attributes unrelated to clothing. We show how the inputs and outputs are fed and generated during inference through Fig. 11.

We use Flow Match Euler Discrete Scheduler as the sampler with 50 sampling steps and a guidance scale of 5.0. The shift value is set to 12.0. We use the Euler solver, and the time discretization follows the scheduler with $\mu = 1$.

Our system runs on a single NVIDIA RTX A6000 GPU. Generating an 81-frame video requires approximately 11 GB of VRAM, and the total inference time, including model loading, is around three minutes per 81-frame sequence.

B.5. Ablation Study Details

For the **Single Module** variant, we remove the GTM and instead attach the garment tokens directly to the HAM inputs. More specifically, all conditioning latents are concatenated frame-wise along the temporal dimension. To address the temporal mismatch, we concatenate the human, garment, and

Table 5. **Quantitative Comparsion** on extended test-set & Additional metrics. **Bold text** indicates the best score within each column.

	ViViD Dataset							Internet Dataset		
Method	$L_1 \downarrow$	PSNR↑	SSIM↑	LPIPS↓	FID↓	$\text{VFID}_{\text{I3D}} \downarrow$	$\text{VFID}_{\text{ResNeXt}} \downarrow$	ID↑	GA↑	AKD↓
VACE (1-stage)	0.1668	12.27	0.5625	0.5584	135.04	23.13	3.34	0.1934	0.6113	5.17
OmniTry → StableAnimator	0.1660	12.24	**0.6807**	0.4650	128.47	20.08	0.59	0.1597	0.5944	19.61
StableAnimator → Magic Try-On	0.1451	12.83	0.6145	0.4576	130.40	19.48	0.90	0.1747	0.5634	19.51
Ours	**0.1144**	**13.94**	0.6773	**0.3662**	**99.73**	**18.07**	**0.36**	**0.3349**	**0.6186**	**3.75**

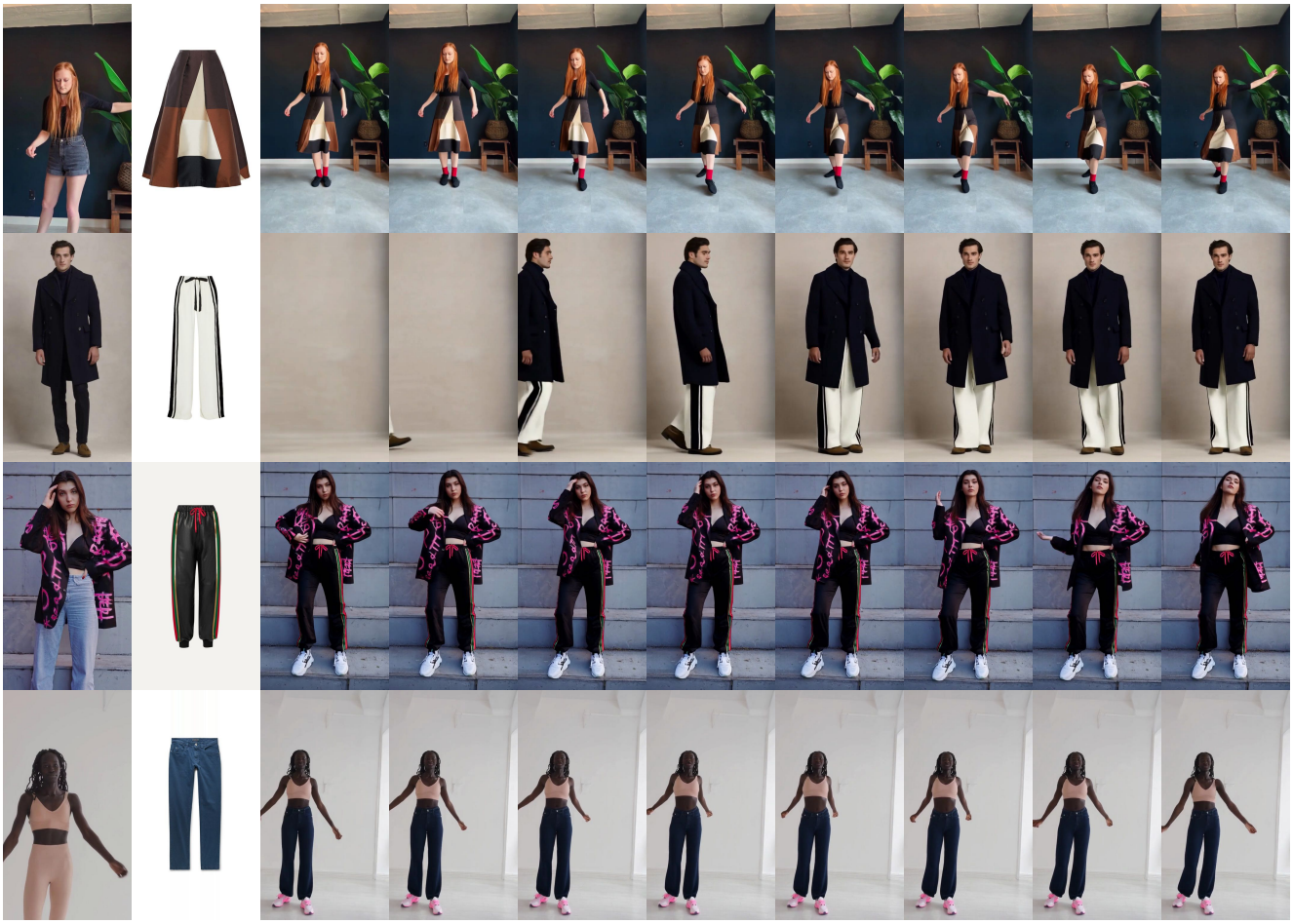

Figure 14. **Additional Results of Lower-Garment Transfer.** We present virtual lower-garment try-on with human image animation results.

pose tokens along the time axis, following the same strategy as in Line 370, i.e., `torch.cat([Human token, Garment token, Pose token], dim=time)`.

For the **Backbone-LoRA** variant, we use a LoRA configuration with rank 128, dropout 0, scaling 0.5, and no rank dropout, resulting in approximately 0.2B trainable parameters. The LoRA layers are applied to the self-attention, cross-attention, and FFN modules in every backbone block. To inject the conditions into the backbone, the noise and pose latents are first concatenated along the channel dimension and then converted into tokens. The human and garment latents are concatenated with zero latents to match the dimensionality of the noise+pose input, tokenized, and then concatenated with the noise+pose tokens.

For the **w/o synthhuman** variant, $\mathbf{I}^G$ is selected using the same strategy described in Sec. 3.1 of main paper. Other than using $\mathbf{I}^G$ as an input, the triplet construction remains unchanged. Accordingly, the training input for this variant is $\mathbf{G}, \mathbf{I}^G, \mathbf{K}, \mathbf{T}$.

Figure 15. **Additional Results of Dress Transfer.** We present virtual dress try-on with human image animation results.

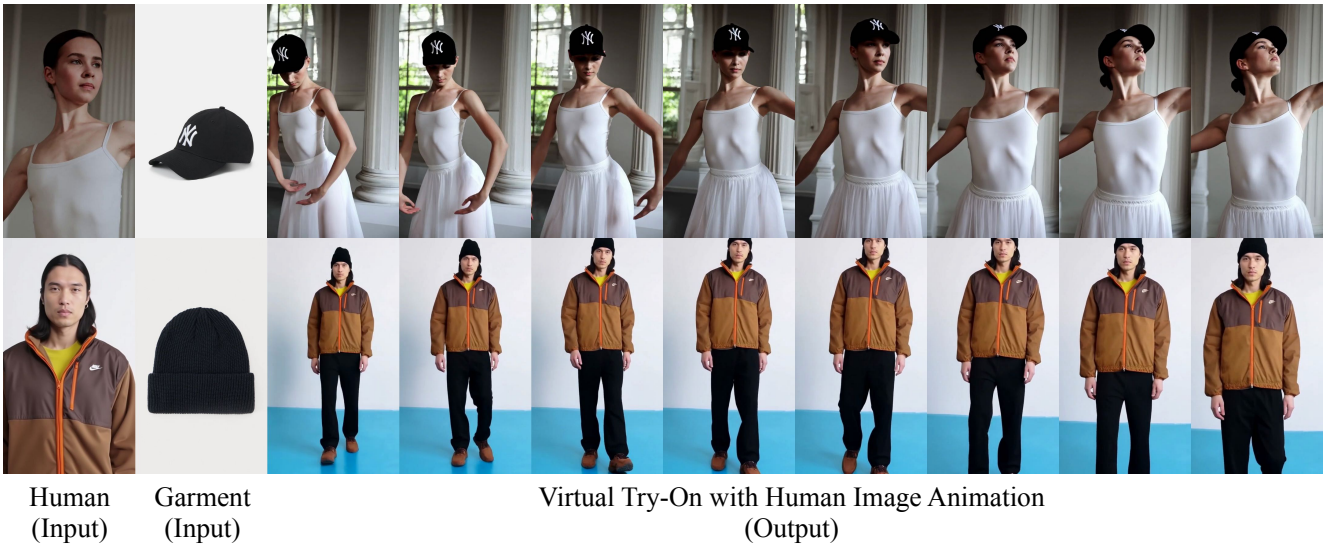

Figure 16. **Additional Results of Hat Transfer.** We present virtual hat try-on with human image animation results.

C. Additional Comparisons and Results

Additional Results. We present additional results of single garment transfer in Fig. 13, Fig. 14, and Fig. 15. Hat transfer results are shown in Fig. 16. As an application, Fig. 17 and Fig. 18 demonstrate that our model supports multiple garment transfer and garment interpolation.

Additional Qualitative Comparisons. We provide additional qualitative comparison results in Fig. 20 and Fig. 21.

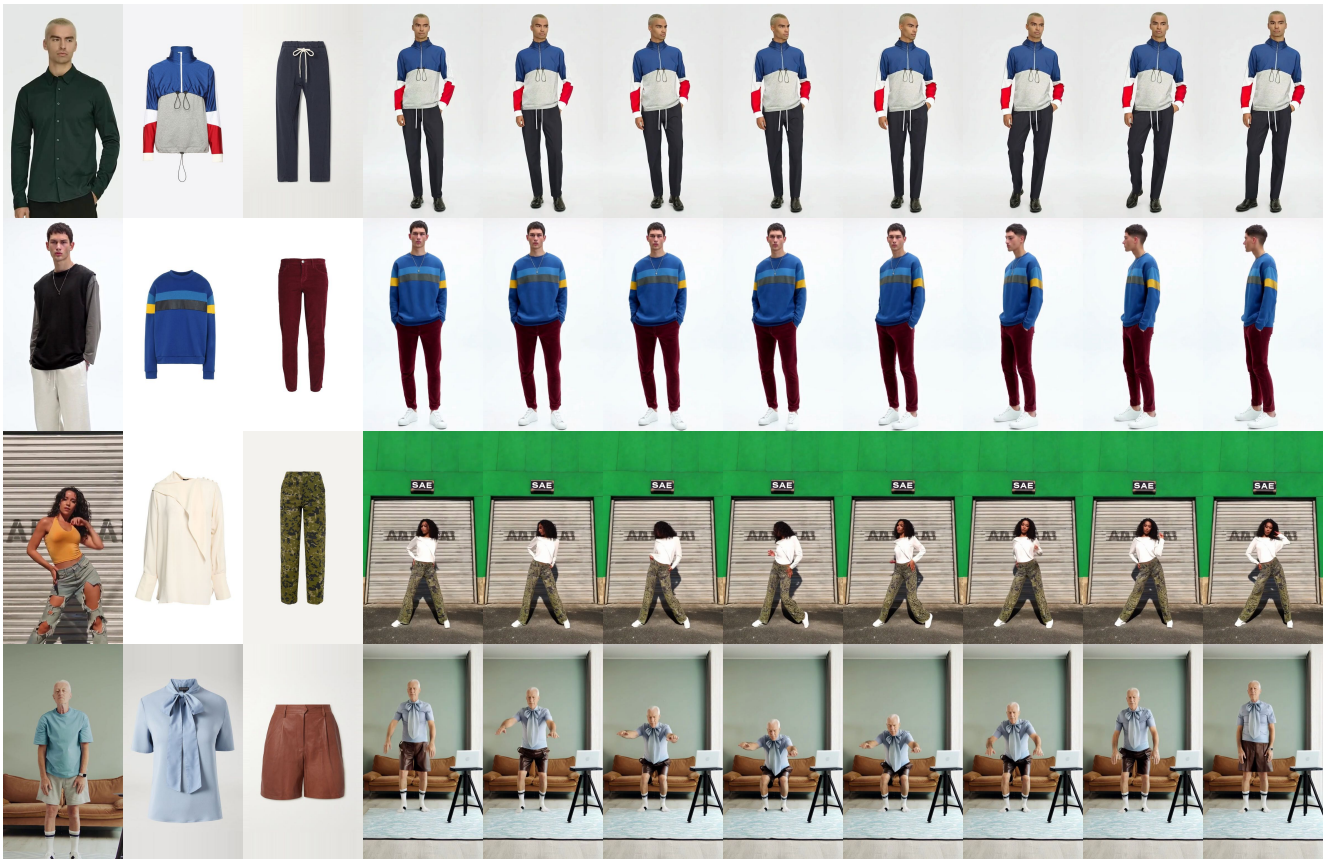

Figure 17. **Additional Results of Multiple-Garment Transfer.** We present virtual multiple garment try-on with human image animation results.

Additional Evaluation. We conduct additional experiments by comparing our method against the three strongest-performing baselines selected from Tabs. 1, 2, and 4. The results are summarized in Tab. 5. For this comparison, we construct a larger test set by randomly sampling 210 videos from the ViViD [5] test split that were not included in our original evaluation. As shown in Tab. 5, our method achieves state-of-the-art performance over all compared baselines on this expanded benchmark.

In addition, we report new quantitative metrics in Tab. 5, including: (1) Identity Preservation (ID), computed using ArcFace [3]; (2) Garment Accuracy (GA), computed using DINOv2 [11]; and (3) Average Keypoint Distance (AKD), following the metric adopted in MRAA [12]. These metrics are evaluated on the Internet dataset used in the main paper, since ViViD does not contain sufficiently visible faces for reliable identity-based evaluation. Our method consistently outperforms the baselines under these additional metrics as well. For temporal consistency, we use VFID as in the original paper and provide the corresponding additional results in Tab. 5.

Figure 18. **Additional Results of In-the-wild Garment Transfer.** We present virtual in-the-wild garment try-on with human image animation results.

Figure 19. **Qualitative Comparisons (Video Virtual Try-On-based).** We compare our results with baselines constructed by combining human animation and video virtual try-on models. We present qualitative comparison results using the ViViD dataset and an Internet dataset. Our method produces the most accurate pose following and garment transfer while preserving identity with high fidelity.

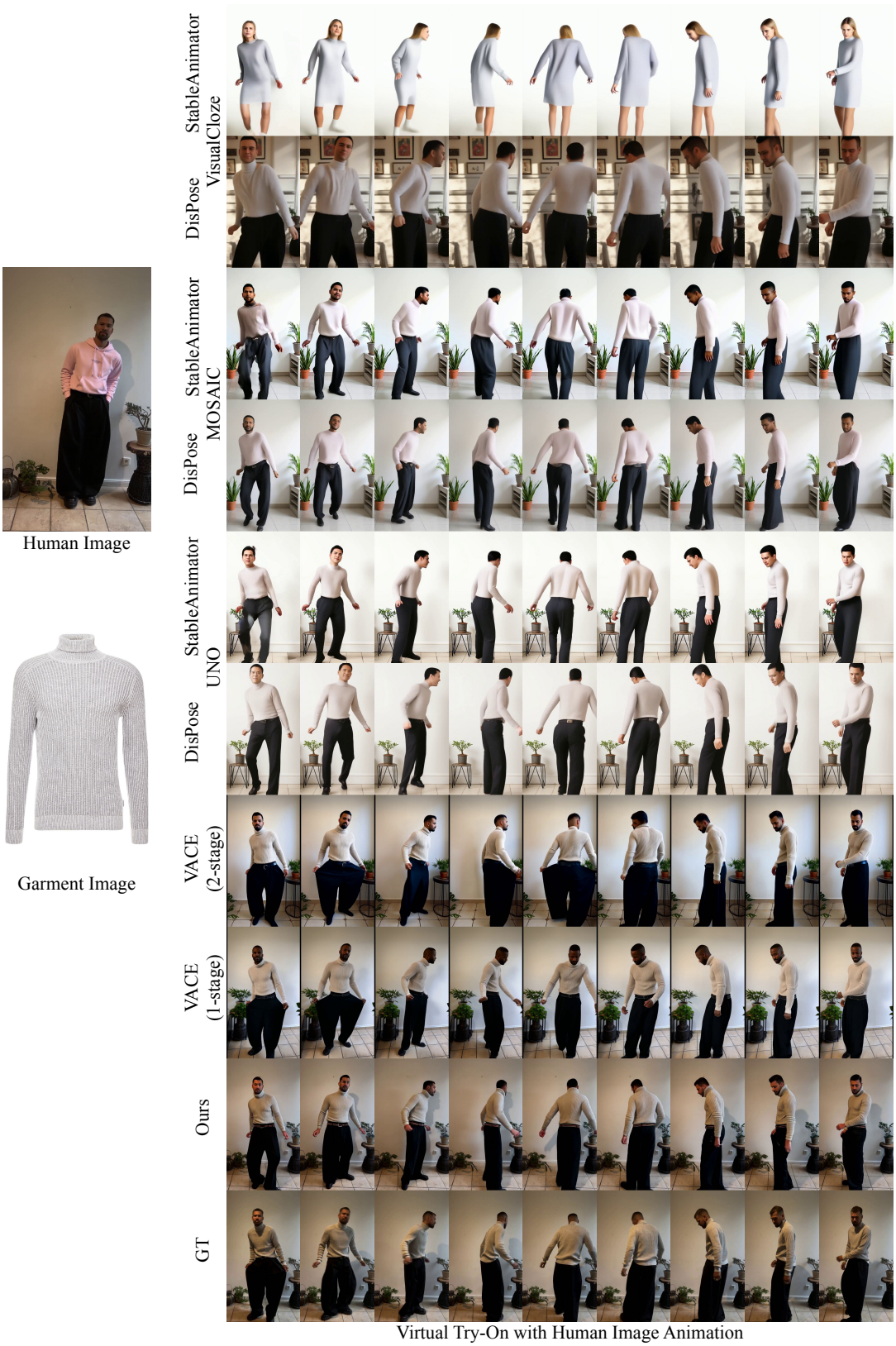

Figure 20. **Additional Qualitative Comparisons (Subject-to-Image-based).** We compare our results with baselines constructed by combining subject-to-image generation and animation models. We present *additional* qualitative comparison over subject-to-image baselines. Our method produces the most accurate pose following and garment transfer while preserving identity with high fidelity.

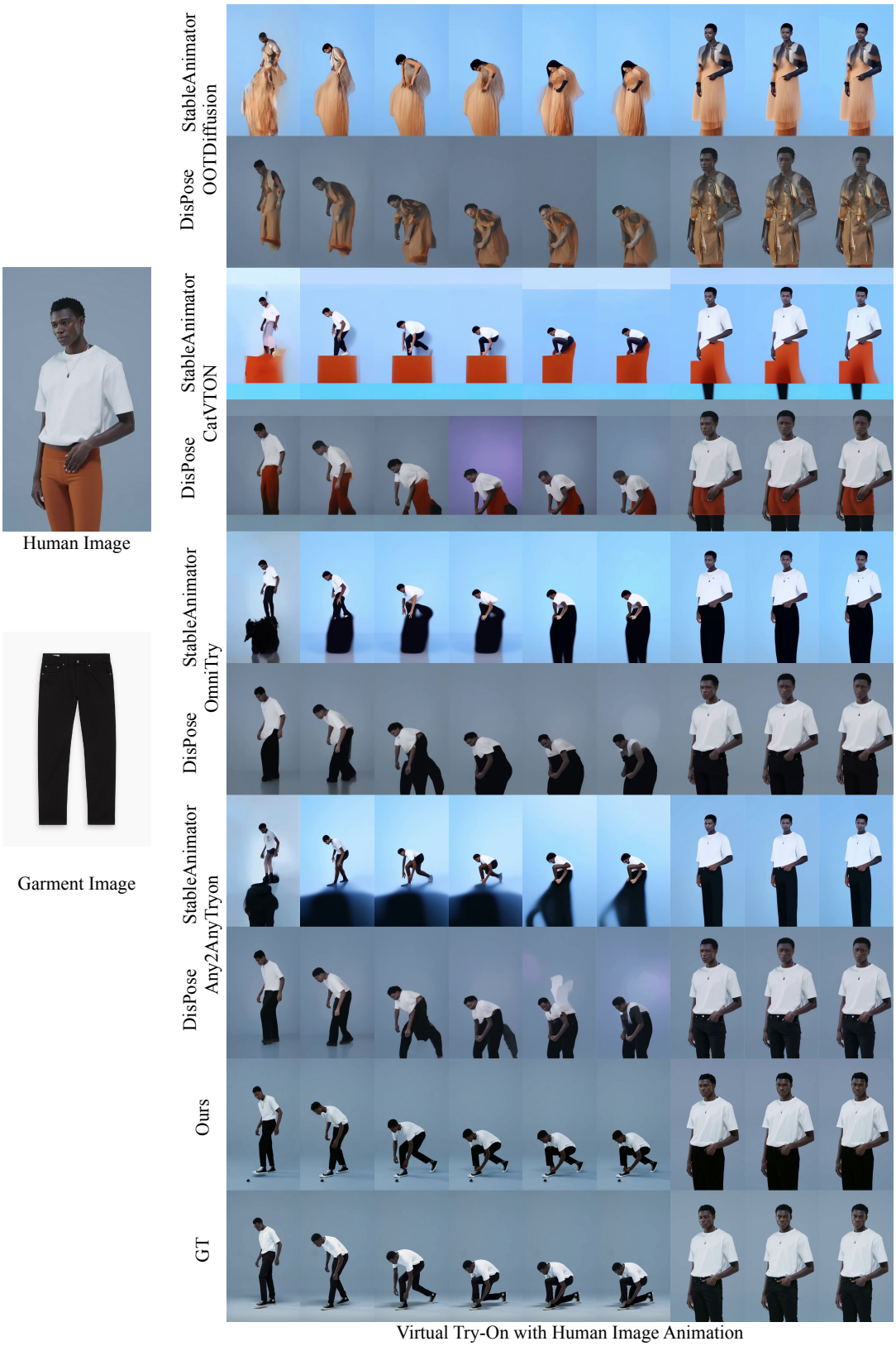

Figure 21. **Additional Qualitative Comparisons (Image Virtual Try-On-based).** We compare our results with baselines constructed by combining image virtual try-on models with animation models. We present *additional* qualitative comparison over image virtual try-on baselines. Our method produces the most accurate pose following and garment transfer while preserving identity with high fidelity.

References

[1] Z. Chen, W. Wang, Y. Cao, Y. Liu, Z. Gao, E. Cui, J. Zhu, S. Ye, H. Tian, Z. Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024. 3

[2] H. W. Chung, N. Constant, X. Garcia, A. Roberts, Y. Tay, S. Narang, and O. Firat. Unimax: Fairer and more effective language sampling for large-scale multilingual pretraining. *arXiv preprint arXiv:2304.09151*, 2023. 2

[3] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019. 6

[4] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Proc. ICML*, 2024. 3

[5] Z. Fang, W. Zhai, A. Su, H. Song, K. Zhu, M. Wang, Y. Chen, Z. Liu, Y. Cao, and Z.-J. Zha. Vivid: Video virtual try-on using diffusion models. *arXiv preprint arXiv:2405.11794*, 2024. 1, 6

[6] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 3

[7] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 2017. 1

[8] Z. Jiang, Z. Han, C. Mao, J. Zhang, Y. Pan, and Y. Liu. Vace: All-in-one video creation and editing. *arXiv preprint arXiv:2503.07598*, 2025. 1, 2

[9] G. Li, S. Zheng, H. Zhang, J. Chen, J. Luan, B. Ou, L. Zhao, B. Li, and P.-T. Jiang. Magictryon: Harnessing diffusion transformer for garment-preserving video virtual try-on. *arXiv preprint arXiv:2505.21325*, 2025. 1

[10] H. Nguyen, Q. Q.-V. Nguyen, K. Nguyen, and R. Nguyen. Swifttry: Fast and consistent video virtual try-on with diffusion models. In *AAAI*, 2025. 1

[11] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 6

[12] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe. First order motion model for image animation. *NeurIPS*, 2019. 6

[13] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024. 1

[14] S. Tu, Z. Xing, X. Han, Z.-Q. Cheng, Q. Dai, C. Luo, and Z. Wu. Stableanimator: High-quality identity-preserving human image animation. In *CVPR*, 2025. 1

[15] T. Wan, A. Wang, B. Ai, B. Wen, C. Mao, C.-W. Xie, D. Chen, F. Yu, H. Zhao, J. Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 2