Supplementary Material

A Architecture Details

We show our overall video diffusion model (VDM) architecture in Figure 2. Our model is trained following v-prediction (Salimans & Ho, 2022), where the model outputs the predicted noise ϵ'_t at timestep t, such that $z'_{t-1} = z_t + \epsilon'_t$. The $L2$-loss is computed in ϵ space.

Input Preprocessing: The conditional inputs to our VDM are 1 or 3 segmented garment images I_g, their corresponding 2D poses J_g, and a sequence of driving 2D and 3D poses (J_{2D}, J_{3D}). In the case of multiple input garment images, I_g is channel-wise concatenation of the segmented garment images. Additionally, I_g and its corresponding spatially-aligned 2D poses J_g^{2D} are concatenated channel-wise.

Conditioning Inputs: The noisy video z_t, garment signals $[I_g, J_g]$, and driving 2D poses J_{2D} are encoded by separate UNet encoders (Zhu et al., 2023) into features f_z, f_g, f_{j2D}, respectively. The driving 3D poses J_{3D} are separately encoded by 4 dense layers into features f_{j3D} and reshaped to match f_{j2D}. The conditioning input embeddings f_g, f_{j2d}, f_{j3d} are processed by the DiT blocks (Peebles & Xie, 2022) before the UNet decoder. We concatenate the driving pose features, f_{j2D} and f_{j3d}, with the noisy video features f_z. The input garment image and pose features f_g are cross-attended with noisy video features f_z, in order to implicitly warp the input garment features to their target locations according to the driving poses (Zhu et al., 2023), getting warped features f'_z.

UNet: Similarly to Fashion-VDM (Karras et al., 2024), we add 3D convolution, temporal attention, and temporal mixing blocks after the two lowest-resolution spatial layers in the UNet encoder and decoder. However, unlike Fashion-VDM, we duplicate these temporal blocks, such that one set only processes video batches and the other only processes 3D spin batches. We implement this switch in the network via conditional network branching. Note that only one branch of temporal blocks is activated at a time. Finally, a UNet decoder decodes f'_z into the predicted noise ϵ_t.

Additional implementation details of our architecture include:

- The kernel sizees of our Conv2D and Conv3D blocks are $(3, 3)$ and $(4, 3, 3)$, respectively.
- Our 8 DiT blocks are implemented with 8 attention heads each and a hidden size of 2048 at feature resolution 32x24.
- Our UNet encoders consist of 4 downsampling CNN blocks. Symmetrically, our UNet decoders consist of 4 upsampling CNN blocks.

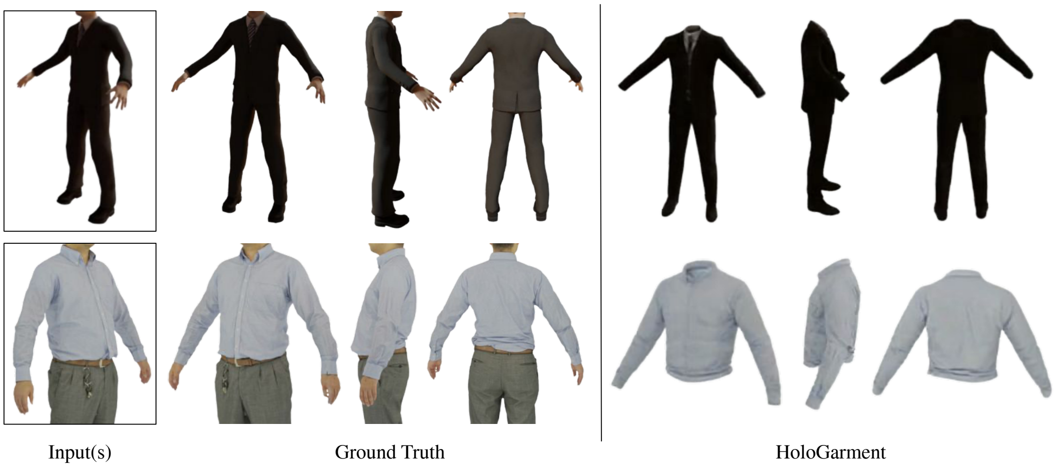

Figure 8: **Qualitative Results on Synthetic 3D Garments.** HoloGarment generates consistent 360-degree novel views from a single garment view that retain high-fidelity to their corresponding ground-truth synthetic 3D assets.

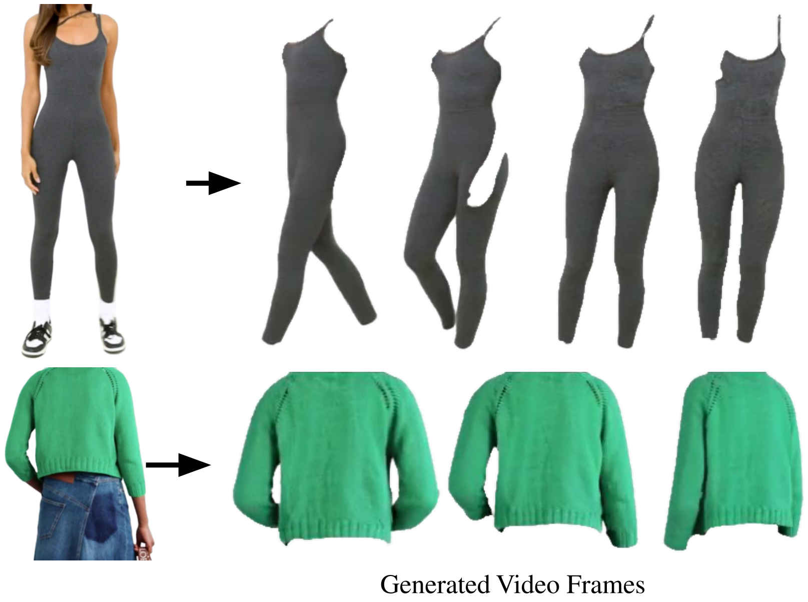

Figure 9: **Qualitative Animation Results.** HoloGarment generates realistic garment animations given a garment image and dynamic driving pose sequence.

B Training and Inference Details

Training: We pretrain our base model without temporal layers on our custom garment image dataset for 1M iterations with batch size 8 (Karras et al., 2024). Then, we train our full model with temporal layers for 589K iterations, approximately 3 days, on 16 TPU-v4's. In this stage of training, we train jointly with 33% fashion image data, 33% fashion video data, and 33% novel views rendered from 3D garment assets, each with frame count 32 and resolution 512 x 384. For task (2) batches, 3D garment assets are rendered as 360-degree RGB spin videos, similar to (Gao et al., 2024). However, different from (Gao et al., 2024), which uses camera position as conditioning, we compute J_{2D} and J_{3D} for both garment spins and real fashion videos, so that the motion representation is shared. Plus, 2D and 3D together encapsulate camera pose (Lu, 2018). Finally, we finetune the model for an additional 50K iterations on 3D data only.

For both pretraining and training, we use an Adam optimizer with linearly decaying learning rate of $1e{-}4$ to $1e{-}5$ over a maximum of 1M iterations. We additionally add independent dropout for each conditional input with 10% probability per batch. After each forward pass, we compute the L2 loss on predicted noise ϵ_t at diffusion timestep t.

Inference: During inference, we use the DDPM sampler (Ho et al., 2020) with 1000 refinement steps to generate 32-frame videos. For evaluating our full and ablated models, we employ dual classifier-free guidance (Brooks et al., 2023) with conditioning groups ($\emptyset$, I_g, $[J_{2D}, J_{3D}]$) and weights $(1, 5, 1)$.

C Comparisons to State-of-the-Art Details

Gemini 2.5 Flash Image: We compare our method to Gemini 2.5 Flash Image (Google, 2025) on garment novel view synthesis. For quantitative evaluation, we generate the front view of each input segmented garment using the prompt *"Generate a front-facing image of the garment in a-pose and without occlusions."* As Gemini 2.5 Flash Image is not designed for temporal consistency, we omit FVD in the quantitative comparison. For qualitative comparisons, we specify the target output angle of the garment in the prompt. Despite synthesizing high-quality, plausible novel views, Gemini 2.5 Flash Image is not designed for temporally-consistency, and struggles to generate smooth 360° novel views.

Veo3: We qualitatively compare with Veo3 (Google DeepMind, 2024) frame-to-video functionality through the Flow app of Google labs using the input segmented garment image as the input frame and the prompt *"Generate a 360-degree orbit of this garment in a-pose and without occlusions."* Veo3 generates high-quality, consistent orbits, but over-saturates the garment colors.

Stable Virtual Camera: We evaluate the official implementation of Stable Virtual Camera (SVC) (Zhou et al., 2025) provided by the authors. We run SVC in single-image video generation mode on the input segmented garment image, following an orbit trajectory for 32 target frames and using all other default parameters. In Figure 5, SVC fails to inpaint occluded regions (bottom row), synthesize plausible novel views (top row) or generate the garment in a canonical a-pose.

Garment3DGen: We follow the official Garment3DGen implementation. Garment3DGen does not provide a texturing tool, so we use a text-to-texture model (Deng et al., 2025), as suggested by the authors. We caption each input image using Gemini (Team, 2023), then use FlashTex (Deng et al., 2025) to add texture to the 3D mesh. From the caption, we also determine the garment type and select the closest template mesh provided by the authors. As shown in Figure 5, the requirement of a template mesh severely limits Garment3DGen's ability to generalize to different garment shapes. Plus, the generated textures show poor fidelity to the input garment image. In contrast, HoloGarment is independent of any template meshes or third-party texture generation methods, making it robust to diverse garment shapes and textures.

CAT3D: For comparisons with CAT3D, we follow the original authors' single-image implementation. Figure 5 shows that CAT3D generates a flattened appearance in side and back views. Additionally, as shown in the bottom row, CAT3D is not robust to occlusions or pose variations, reproducing the input holes and wrinkling. Finally, unlike our method, CAT3D cannot warp the garment into a desired target pose, replicating the original garment pose.

D Additional Qualitative Results

We show additional qualitative image-to-3D, sparse view-to-3D, and video-to-3D results of our method in Figure 11. We also show qualitative results of our single-view method on held-out synthetic 3D garment assets in Figure 8. Given a single view of a synthetic 3D asset, HoloGarment synthesizes plausible novel views that are consistent and retain high fidelity to the ground truth views.

E Garment Animation

In Figure 9, we demonstrate HoloGarment's ability to realistically animate real-world garments given an image and driving pose sequence (task 1). Due to the nature of the real-world image and video data, HoloGarment creates wrinkling and occlusions when operating on video data. Although the focus of our work is NVS, the ability for HoloGarment to perform well on the image animation task, is crucial for enabling video-to-NVS finetuning (Section 5.4).

F Failure Cases

In Figure 10, we show two examples of failure cases of our method. These include hallucinating bottoms for top-only garment inputs and at times misrepresenting skirts as pants, especially when a slit is present in the input view.

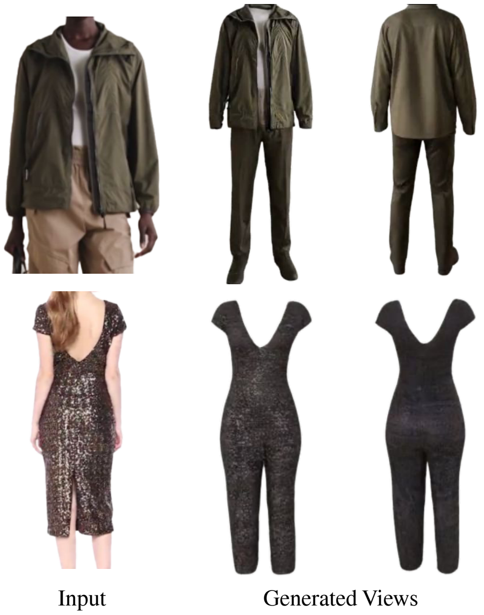

Figure 10: **Failure Cases.** At times, HoloGarment hallucinates pants when the input garment is top-only (*left*). In other cases, due to ambiguities in the input (e.g., dress slits, extremely large occlusions), it may generate an incorrect garment shape (*right*).

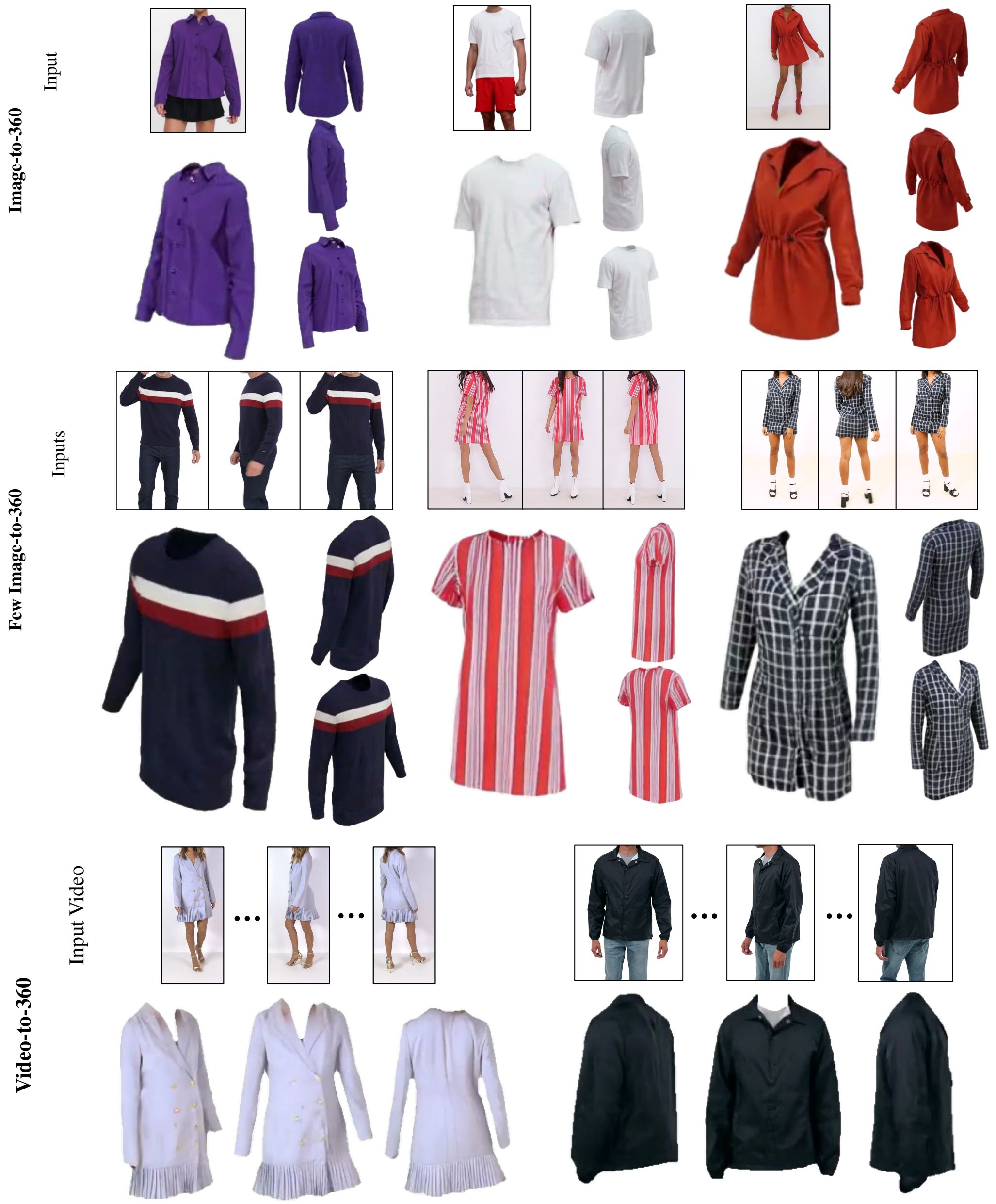

Figure 11: **Additional Qualitative Results.** Demonstrating our method's capability for 360° novel view synthesis of garments from a 1-3 input image(s) or a video. Results are best viewed in our supplementary video.