## Contents



---

## A. Supplementary Video

To better demonstrate the efficacy of our framework and the visual quality of our results, we provide a comprehensive supplementary video (overall length 2′47′′). We strongly recommend viewing the video to fully dynamic visual results.

## B. Proof

**Proposition B.1 (Error Bound of Gradient Approximation)** *Consider the score approximation* $\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) \approx \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\hat{\mathbf{x}}_{0|t})$ *used in Eq.(10). Let* $\mathcal{M}$ *be the measurement operator and* $\hat{\mathbf{x}}_{0|t} = \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]$ *be the posterior mean. Under the manifold constraint, the approximation error* $\epsilon$ *is upper bounded by:*

$$\epsilon \leq C \cdot \|\mathcal{M}\|_2 \cdot \mathbb{E}_{\mathbf{x}_0 \sim p(\mathbf{x}_0|\mathbf{x}_t)}[\|\mathbf{x}_0 - \hat{\mathbf{x}}_{0|t}\|], \tag{12}$$

*where* $C$ *is a constant related to the Lipschitz property of the noise schedule.*

**Proof:** Following the theoretical framework in DPS [8], the likelihood gradient can be decomposed via the Tweedie's formula. The spectral norm $\|\mathcal{M}\|_2$ represents the maximum amplification factor of the measurement operator.

In our specific task, the operator $\mathcal{M}$ is defined as a binary mask $\mathbf{M} \in \{0, 1\}^n$. The spectral norm of a diagonal matrix (or masking operator) is given by its maximum singular value:

$$\|\mathcal{M}\|_2 = \max_i |M_{ii}| = 1. \tag{13}$$

Consequently, the error bound simplifies to $\epsilon \leq C \cdot \mathbb{E}[\|\mathbf{x}_0 - \hat{\mathbf{x}}_{0|t}\|]$. This term represents the uncertainty of the posterior estimation at time $t$. As the diffusion process approaches the clean data manifold ($t \to 0$), the posterior distribution $p(\mathbf{x}_0|\mathbf{x}_t)$ collapses to a Dirac delta distribution $\delta(\mathbf{x}_0 - \hat{\mathbf{x}}_0)$, leading to $\epsilon \to 0$. This ensures that the approximate gradient converges to the true score direction in the final sampling stages. □

**Proposition B.2 (SDE Correction Mechanism [24])** *The continuous implicit Langevin diffusion $d\boldsymbol{x}_t = \frac{1}{2}\nabla \log p(\boldsymbol{x})dt + d\mathbf{w}_t$ actively corrects sampling errors by admitting the data marginal $p(\boldsymbol{x})$ as its unique stationary distribution.*

**Proof:** The time evolution of the probability density $p_t(\boldsymbol{x})$ is governed by the Fokker-Planck Equation (FPE):

$$\frac{\partial p_t}{\partial t} = -\nabla \cdot \left(\frac{1}{2}(\nabla \log p)p_t\right) + \frac{1}{2}\Delta p_t. \tag{14}$$

We verify the stationarity by setting $p_t(\boldsymbol{x}) = p(\boldsymbol{x})$. Using the identity $(\nabla \log p)p = \nabla p$, the drift term becomes $-\frac{1}{2}\nabla\cdot(\nabla p) = -\frac{1}{2}\Delta p$. This exactly cancels the diffusion term $\frac{1}{2}\Delta p$, yielding $\frac{\partial p_t}{\partial t} = 0$. Thus, the dynamics inherently drive any distribution towards $p(\boldsymbol{x})$, correcting deviations accumulated from prior steps. □

**Proposition B.3 (Equivalence of Stochastic Term.)** *Our proposed stochastic sampling step, which acts on the noise prediction component, acts as a valid discretization of a reverse-time SDE by introducing an explicit diffusion term to the standard Rectified Flow ODE.*

**Proof:** Recall that the standard deterministic (ODE) update in Rectified Flow is given by linear interpolation:

$$\boldsymbol{x}_{t_{\text{next}}} = (1 - t_{\text{next}})\hat{\boldsymbol{x}}_{0|t} + t_{\text{next}}\hat{\boldsymbol{x}}_{1|t}. \tag{15}$$

Our method introduces stochasticity by perturbing the target noise prediction $\hat{\boldsymbol{x}}1|t$. Specifically, we replace $\hat{\boldsymbol{x}}_{1|t}$ with $\hat{\boldsymbol{x}}^{\text{stoch}}_{1|t} = \sqrt{1-\gamma}\hat{\boldsymbol{x}}_{1|t} + \sqrt{\gamma}\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ and $\gamma$ is a scheduling parameter. Substituting this into the update rule yields:

$$\boldsymbol{x}^{\text{SDE}}_{t_{\text{next}}} = (1 - t_{\text{next}})\hat{\boldsymbol{x}}_{0|t} + t_{\text{next}}\left(\sqrt{1-\gamma}\hat{\boldsymbol{x}}_{1|t} + \sqrt{\gamma}\boldsymbol{\epsilon}\right) \tag{16}$$

$$= \underbrace{\left[(1 - t_{\text{next}})\hat{\boldsymbol{x}}_{0|t} + t_{\text{next}}\sqrt{1-\gamma}\hat{\boldsymbol{x}}_{1|t}\right]}_{\text{Effective Drift (Deterministic)}} + \underbrace{[t_{\text{next}}\sqrt{\gamma}\boldsymbol{\epsilon}]}_{\text{Effective Diffusion (Stochastic)}}. \tag{17}$$

The resulting update equation takes the form of a standard Euler-Maruyama discretization of an SDE ($d\boldsymbol{x} = \mathbf{f}(\boldsymbol{x}, t)dt + g(t)d\mathbf{w}$). The first term represents the drift (the intended restoration path), while the second term represents the diffusion ($g(t)d\mathbf{w}$), with the noise magnitude scaled by $t_{\text{next}}\sqrt{\gamma}$. This explicitly proves that our method injects the necessary stochasticity to correct out-of-distribution (OOD) errors during sampling. □

**Derivation of closed-form guidance.** To enforce the identity constraint, we minimize the loss $\mathcal{L} = \|\mathcal{M} \odot (\boldsymbol{x} - \hat{\boldsymbol{x}}_{0|t})\|^2$ with respect to the noisy latent $\boldsymbol{x}_t$. Applying the chain rule yields $\nabla_{\boldsymbol{x}_t}\mathcal{L} = (\frac{\partial \hat{\boldsymbol{x}}_{0|t}}{\partial \boldsymbol{x}_t})^\top \nabla_{\hat{\boldsymbol{x}}_{0|t}}\mathcal{L}$. Calculating the exact Jacobian $\frac{\partial \hat{\boldsymbol{x}}_{0|t}}{\partial \boldsymbol{x}_t}$ requires computationally expensive backpropagation through the diffusion backbone. To achieve an efficient closed-form solution, we follow standard Diffusion Posterior Sampling practice and approximate this Jacobian as a scalar identity matrix (absorbing scaling factors into the step size $\lambda(t)$). Consequently, the gradient simplifies directly to the masked residual $\nabla_{\boldsymbol{x}_t}\mathcal{L} \propto -\mathcal{M} \odot (\boldsymbol{x} - \hat{\boldsymbol{x}}_{0|t})$, enabling fast, derivative-free guidance updates.

# C. Background

## C.1. 3D Gaussian Splatting

3D Gaussian Splatting (3D-GS) [25] is a photorealistic 3D scene representation and real-time rendering technique [10]. Instead of using traditional polygons or volumetric grids, 3D-GS models a scene as a collection of millions of explicit, anisotropic 3D Gaussians. Each Gaussian is defined by several key properties: its 3D position (mean), shape (a 3D covariance matrix, allowing it to be a sphere, needle, or flat disk), color (often represented by Spherical Harmonics to capture view-dependent effects), and opacity (alpha). The scene is created by optimizing these properties, typically starting from a sparse point cloud generated by Structure-from-Motion (SfM). During this optimization, a process of adaptive density control dynamically adds (clones) or removes (prunes) Gaussians to efficiently reconstruct fine details. To render a new view, these 3D Gaussians are projected onto the 2D image plane, sorted by depth, and alpha-blended back-to-front in a highly efficient rasterization process.

### C.2. 4D Gaussian Splatting

To extend 3D-GS to dynamic scenes, 4D Gaussian Splatting (4D-GS) [77] techniques model how Gaussians move and change over time. Instead of storing separate 3D-GS models for each frame, a holistic 4D representation is learned. A common strategy is to define a set of canonical 3D Gaussians and then predict their deformation at any given timestamp. To efficiently encode this 4D space-time information, methods often employ a decomposed neural voxel grid, drawing inspiration from HexPlane [5]. This approach factorizes the 4D space $(x, y, z, t)$ into several lower-dimensional planes (e.g., $xy$, $xz$, $yt$). To find a Gaussian's deformation, its 4D coordinates are used to query features from these planes. The aggregated features are then passed through a lightweight MLP to predict the transformation (such as translation or rotation), allowing the scene to be reconstructed at novel times.

### C.3. Video Diffusion Transformer Backbone

Our framework leverages the Wan [72] architecture, a state-of-the-art text-to-video model built upon the Diffusion Transformer [9, 46, 53] (DiT) paradigm. This architecture consists of three core components: 1) A spatio-temporal VAE that compresses input videos from pixel space into a compact latent space; 2) A robust text encoder (e.g., umT5), selected for its multilingual capabilities and convergence properties, to encode text prompts; 3) The central Diffusion Transformer, which processes sequences of video latent tokens. Within the Transformer blocks, text conditions are injected via cross-attention to ensure semantic fidelity. Temporal information is embedded using a shared MLP that predicts modulation parameters for each block, a design that efficiently enhances performance with minimal parameter overhead.

The model is trained using the Flow Matching framework, specifically Rectified Flows (RF), which provides a stable and theoretically grounded generative process. RF models the transition from pure noise $\boldsymbol{x}_0$ to the real data latent $\boldsymbol{x}_1$ as a linear interpolation (Ordinary Differential Equation). For a time step $t \in [0, 1]$, the training input $\boldsymbol{x}_t$ is defined as:

$$\boldsymbol{x}_t = t \cdot \boldsymbol{x}_1 + (1 - t) \cdot \boldsymbol{x}_0. \tag{18}$$

The model is trained to predict the velocity field $\mathbf{v}_t$ of this trajectory, where the ground truth velocity is simply $\mathbf{v}_t = \boldsymbol{x}_1 - \boldsymbol{x}_0$. The training objective minimizes the mean squared error (MSE) between the predicted and ground truth velocity. Training follows a multi-stage curriculum, progressing from low-resolution images to high-resolution joint image-video training.

## D. More Implementation Details

### D.1. Preserved Area Segmentation

To ensure identity preservation, we define a preserved area mask $\mathcal{M}$. We utilize Grounded-DINO-SAM2 to segment the human region, denoted as $\mathcal{M}_{\text{human}}$, and the garment region, $\mathcal{M}_{\text{garment}}$. The final preserved area is obtained by excluding the garment region from the human mask:

$$\mathcal{M} = \mathcal{M}_{\text{human}} \setminus \mathcal{M}_{\text{garment}}. \tag{19}$$

To align with the latent space of the Video VAE, we downsample the binary mask $\mathcal{M}$ to match the latent dimensions (specifically, downsampling by a factor of 8 spatially and 4 temporally).

### D.2. Residual Field Configuration

For the non-rigid motion modeling, we employ a multi-resolution HexPlane module. The base resolution $R(i, j)$ is set to 64 and is progressively upsampled by a factor of 2. The Gaussian deformation decoder is implemented as a lightweight MLP using zero-initialization for the final layer weights, ensuring the deformation field starts as an identity mapping.

### D.3. Personalized Video Diffusion

**Control DiT via Channel-wise Concatenation.** We inject dense spatiotemporal conditions (e.g., the control video) directly into the main branch via latent space augmentation. Unlike adapter-based methods that operate on intermediate features, we concatenate the encoded control latents $\mathbf{y}$ with the noisy video latents $\boldsymbol{x}$ along the channel dimension prior to the patch embedding layer. Formally, the input to the DiT becomes $[\boldsymbol{x}; \mathbf{y}] \in \mathbb{R}^{B \times (C_x + C_y) \times F \times H \times W}$. This strategy ensures that every spatial patch processed by the Transformer is explicitly conditioned on the corresponding local structural information.

**Reference Image Fusion.** To achieve appearance transfer, we treat the reference image as a visual prompt. The reference image is encoded into latents and passed through a projection layer to match the embedding dimension of the DiT. These projected features are flattened and concatenated with the video tokens along the sequence dimension, effectively serving

| Method | Training Objective | Single-view Image Input | Skeleton-controllable | Identity Preservation | Non-rigid Motion | High-quality Rendering |
|---|---|---|---|---|---|---|
| Disco4D [51] | MSE+SDS | ✓ | × | ✓ | × | × |
| AKD [36] | SDS | ✓ | × | ✓ | × | × |
| PhysAvatar [89] | MSE | × | ✓ | ✓ | ✓ | ✓ |
| SV4D/SV4D 2.0 [81, 85] | MSE | ✓ | × | × | ✓ | × |
| CharacterShot [12] | MSE | ✓ | ✓ | × | ✓ | × |
| Human4DiT [63] | MSE | ✓ | ✓ | × | ✓ | × |
| PERSONA [64] | MSE | ✓ | ✓ | × | ✓ | × |
| LHM [56] | - | ✓ | ✓ | ✓ | × | × |
| Ours | MSE | ✓ | ✓ | ✓ | ✓ | ✓ |

Table 3. Difference among the other human (character) animation methods. "-" means there is no optimization process in the animation.

as a "visual prefix." By integrating the reference signal into the input sequence, the DiT utilizes its global self-attention mechanism to attend to reference appearance details across all generated frames. These prefix tokens are masked out during the final video reconstruction.

**Training Details.** To facilitate the personalized video generation, we implement the proposed Wan-Control framework based on the `DiffSynth` library. The model is fine-tuned on a curated subset of the TikTok dataset (available via HuggingFace), comprising approximately 20,000 video clips. For high-fidelity motion guidance, we pre-process all video frames using `DWPose` to extract dense human pose annotations. The training process is conducted on a cluster of $8\times$ NVIDIA RTX A6000 GPUs for approximately 15,000 iterations. We utilize a constant learning rate with a batch size optimized for the GPU memory. For further architectural details and hyper-parameter configurations. We also find some good open-sourced alternative, such as https://huggingface.co/alibaba-pai/Wan2.1-Fun-V1.1-1.3B-Control, and https://huggingface.co/alibaba-pai/Wan2.2-Fun-A14B-Control, as our **video backbone**.

### D.4. Baseline Implementation

We mainly classify our baselines in Tab. 3, selecting several representative methods with official implementation[1] for comparison.

**Disco4D [51].** Due to the unavailability of key components in the official repository, we re-implemented the core algorithm within our own framework. Following the supervision strategy of DreamGaussian4D [60], this method combines Mean Squared Error (MSE) loss from a single-view driving video with Score Distillation Sampling (SDS) guidance from Zero-123 [41]. To ensure a fair comparison, we generated the required driving video using our personalized Wan-based model, conditioned on the front-view skeleton rendering and the reference image. We adopted the SDS implementation directly from the DreamGaussian4D repository.

**SV4D 2.0 [85].** SV4D is a multi-view video diffusion model fine-tuned on Stable Video Diffusion (SVD) using a large-scale 4D dataset filtered from Objaverse. It takes a single-view video as input and outputs synchronized multi-view videos. We utilized the same driving video generated for Disco4D as the input. However, we observe a severe identity shift between the output and input videos. We attribute this to the domain gap, as SV4D is trained primarily on synthetic Objaverse objects rather than realistic human captures.

**PERSONA [64].** We utilize the official implementation of PERSONA. This method employs MimicMotion [88], a pose-driven video diffusion model, to generate synthetic video data which is then used to optimize a canonical 3D Gaussian field and a pose-dependent deformation field. The pipeline relies on an extensive set of off-the-shelf components, including Sapiens [26], DECA, and ResShift. Despite incorporating various regularization terms, such as geometry weighted optimization and multiple monocular normal/depth priors, we find that the method struggles to preserve the fine-grained identity of the subject during complex motions.

**LHM [56].** We use the official implementation of LHM as a representative kinematics-based baseline. LHM effectively reconstructs a human from a single-view image with high-fidelity identity and efficient inference speed. However, as it relies purely on kinematics-based deformation to animate the 3D Gaussians, it fundamentally lacks the ability to model non-rigid dynamics such as clothing deformation. (Note: Our method builds upon this kinematics-based representation, using it as a starting point to learn residual non-rigid motions via video diffusion priors.)

[1] For example, since Human4DiT/CharactorShot is not open-sourced, we choose SV4D as a representative method of reconstructing from MV video.

### D.5. Details of Competitive Sampling Methods

We compare our approach against several representative methods capable of transforming low-quality source inputs into high-quality targets using off-the-shelf diffusion priors. Since some of algorithms are designed from DDPM, we implement them in the context of flow matching with their core ideas.

**Vanilla SDEdit [48].** SDEdit serves as the foundational baseline for image and video restoration. The method follows a strictly stochastic process: it first perturbs the source input $\boldsymbol{x}_{\text{src}}$ by adding Gaussian noise to reach an intermediate time step $t_0 \in (0, 1)$. This forward diffusion process effectively destroys high-frequency artifacts. Subsequently, the standard reverse ODE/SDE sampling is applied from $t_0$ to $t = 0$ to generate the restored output. While effective for minor denoising, it often faces a trade-off between preserving identity (low $t_0$) and removing significant artifacts (high $t_0$).

**MCS [74].** Multiview Consistency Sampling (MCS) was originally proposed to balance fidelity and generation quality in 3D scene generation. The authors observe that while higher noise injection improves realism, it degrades the structural fidelity to the input. To mitigate this, MCS modifies the posterior mean during sampling to explicitly include signal from the input image. In our implementation, we adapt this to the Flow Matching framework. At each sampling step, we modify the predicted posterior mean $\hat{\boldsymbol{x}}_{0|t}$ to incorporate a weighted component of the source input $\boldsymbol{x}_{\text{src}}$. This bias term forces the generation trajectory to remain structurally close to the input video, ensuring that the "hallucinated" details align with the original identity.

**HFS-SDEdit [62].** HFS-SDEdit aims to preserve structural details by explicitly fusing frequency components in the latent space. It operates on the hypothesis that the structural identity resides in high-frequency signals. During the reverse sampling process, the method replaces the high-frequency component of the current denoised latent $\boldsymbol{x}_t$ with that of the noisy source input. The update rule is defined as:

$$\boldsymbol{x}'_t = \text{LPF}(\boldsymbol{x}_t) + \text{HPF}(\boldsymbol{x}_{\text{src},t}), \tag{20}$$

where LPF and HPF denote Gaussian low-pass and high-pass filters, respectively, and $\boldsymbol{x}_{\text{src},t}$ is the noised version of input data corresponding to time $t$. This operation forces the solver to generate realistic low-frequency content (lighting, materials) while rigorously adhering to the edges and boundaries of the source input.

**NC-SDEdit [83].** We adapt the Noise Calibration (NC) strategy to our Flow Matching framework. While the original implementation calibrates the noise estimate $\boldsymbol{\epsilon}$, our adaptation operates directly on the estimated clean data (posterior) to ensure structural consistency. In each sampling step $t$, we first solve the flow equation to estimate the clean target $\hat{\boldsymbol{x}}_{0|t}$ from the current noisy latent $\boldsymbol{x}_t$ and predicted velocity $\boldsymbol{v}_t$. We then calibrate this posterior by replacing its high-frequency components with those of the source reference $\boldsymbol{x}_{\text{src}}$:

$$\hat{\boldsymbol{x}}'_{0|t} = \hat{\boldsymbol{x}}_{0|t} - \text{HPF}(\hat{\boldsymbol{x}}_{0|t}) + \text{HPF}(\boldsymbol{x}_{\text{src}}), \tag{21}$$

where $\text{HPF}(\cdot)$ extracts high-frequency details via Fourier transform. Finally, the solver (e.g., Euler step) computes the latent for the next timestep $\boldsymbol{x}_{t_{\text{next}}}$ using this calibrated target $\hat{\boldsymbol{x}}'_{0|t}$. This approach enforces strict structural alignment with the input video throughout the generation trajectory while allowing the low-frequency content to be refined by the diffusion prior.

**FlowEdit [31].** FlowEdit constructs a mapping between source and target distributions by leveraging the reversibility of ODEs. It defines the editing direction based on the difference between a source velocity (conditioned on a source prompt) and a target velocity (conditioned on a target prompt). In our experiments, we utilize a negative prompt (describing low-quality attributes) to match the source distribution and a positive prompt for the target. However, we find that because FlowEdit relies on the model's semantic understanding of the prompt to model the degradation, it often fails to correct the severe, non-semantic out-of-distribution (OOD) artifacts present in the coarse 3D renderings, as these specific artifacts are not easily described by text.

## E. Ablation Studies (Extended)

### E.1. Quantitative Ablation

Table 4 provides a comprehensive quantitative evaluation of each key component within our framework, including the stochastic sampling mechanism, the self-guidance strategy, and the personalized video diffusion module. According to the results, our full model configuration achieves the optimal balance between visual fidelity and identity preservation. Specifically, while the exclusion of self-guidance leads to a marginal improvement in the Frechet Inception Distance, it incurs a substantial degradation in identity consistency, as reflected by the significant drop in the CLIP-Identity score. This observation validates that self-guidance is indispensable for maintaining the subject's unique features throughout the generation

process. Furthermore, the integration of stochastic sampling and personalized diffusion proves essential for temporal coherence and motion realism, with the full model yielding the lowest Frechet Video Distance. Although individual modules may favor specific metrics, the synergistic effect of all components ensures that the model produces high-quality videos without compromising the structural or stylistic integrity of the personalized target.

Table 4. Quantitative ablation study of the proposed components. The results demonstrate that the full model configuration achieves the most robust performance across all evaluation metrics.

| Metrics | Coarse Model | w/o Stochastic | w/o Self-Guidance | w/o Personalized | **Full Model** |
|---|---|---|---|---|---|
| FID ↓ | 199.1 | 187.4 | **104.1** | 125.3 | <u>105.3</u> |
| FVD ↓ | 367.0 | 349.7 | <u>298.8</u> | 301.4 | **295.2** |
| CLIP-Identity ↑ | **0.8847** | 0.8804 | 0.8220 | 0.8645 | <u>0.8838</u> |

## E.2. More Results of Self-guided Stochastic Sampling

Due to space constraints in the main manuscript, we provide additional qualitative comparisons to validate the efficacy of our core technical contribution: **self-guided stochastic sampling**. We evaluate our approach against two distinct baselines: 1) **Direct Generation**, which uses the pretrained video model directly (with reference image and 2D skeleton sequence); and 2) **Standard ODE-based Restoration**, where we employ MCS [74] as a representative deterministic sampling method. Dynamic visualizations of these comparisons can be found in the Supplementary Video (`00:40 - 01:32`).

As illustrated in Fig. 13, the challenges of this task are evident. The initial **mesh-rigged animation** (Input) exhibits significant artifacts, including unnatural garment dynamics and blurred edges, consistent with the limitations discussed in the main paper (e.g., Fig. 1). **Direct Generation**, while achieving high realism, suffers from severe identity loss and hallucinations; notably, the model generates extraneous accessories such as a bag (Row 2) or a watch (Row 3), rendering it unsuitable for faithful reconstruction. Furthermore, standard **ODE-based sampling** (MCS) fails to effectively correct the out-of-distribution nature of the coarse rendering, resulting in over-smoothed textures and persistent blurring along garment boundaries. In contrast, our **self-guided stochastic sampling** effectively bridges the gap between realism and fidelity. It restores photorealistic details and valid non-rigid dynamics while strictly preserving the original human identity.

## E.3. Sensitivity Analysis of Initial Noise Strength

The initial noise strength, denoted as $t_0$, serves as the critical hyperparameter in our self-guided stochastic sampling strategy. It governs the trade-off between the restoration capability and the fidelity to the initial coarse rendering. As illustrated in Fig. 10, we conduct a comprehensive sensitivity analysis by varying $t_0$ across the range $[0.2, 0.8]$. At lower noise levels ($t_0 \in \{0.2, 0.4\}$), the sampling trajectory is too short to effectively correct the Out-of-Distribution (OOD) artifacts, resulting in outputs that retain the degradation of the source mesh-rigged animation. Conversely, at higher noise levels ($t_0 \in \{0.6, 0.8\}$), our method demonstrates significant robustness. Unlike standard restoration methods where high noise often leads to identity loss, our self-guidance mechanism ensures that the subject's identity remains remarkably stable even at $t_0 = 0.8$. Ultimately, we empirically select $t_0 = 0.6$ as the default setting, as it strikes an optimal balance between generation quality, identity preservation, and sampling efficiency.

## E.4. More Ablations in 4D Optimization

**Adaptive densification.** Adaptive densification and pruning are fundamental mechanisms in 3D Gaussian Splatting for capturing high-frequency details. We incorporate these strategies into our photorealistic 4D reconstruction pipeline. As demonstrated in Fig. 11a, relying solely on the deformation of the canonical geometry is insufficient to model complex texture dynamics (e.g., shifting wrinkles). Without densification, the model fails to allocate sufficient primitives to these dynamic regions, causing the clothing textures to appear significantly blurred.

**Mask loss regularization.** Prior works, such as PERSONA, have established that geometric constraints are critical for fidelity. We validate this by ablating the mask loss during our 4D optimization. This regularization is particularly important in conjunction with our densification strategy. As shown in Fig. 11b, without the mask loss to constrain the generation of new primitives, "floaters" emerge in free space, and the boundary definition of the subject degrades compared to our full setting.

**Iterative dataset update.** We compare our iterative dataset update strategy against a standard single-stage optimization. As illustrated in Fig. 11c, single-stage optimization tends to result in over-smoothed textures, effectively "averaging out" high-frequency details due to inherent view and temporal inconsistencies in the initial supervision. In contrast, employing

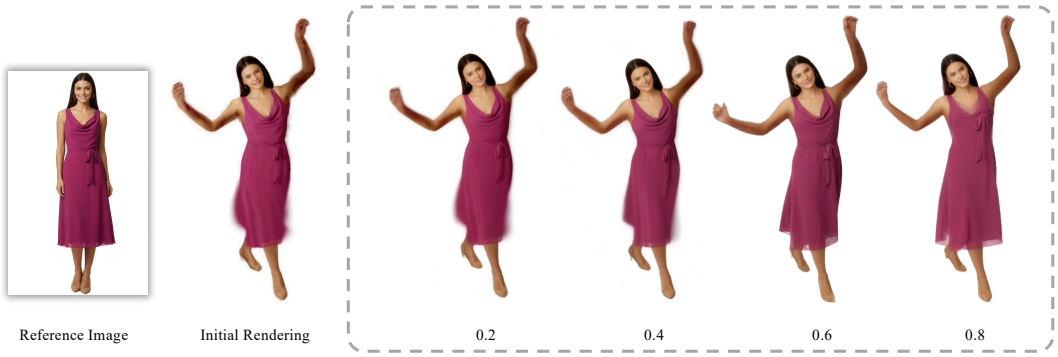
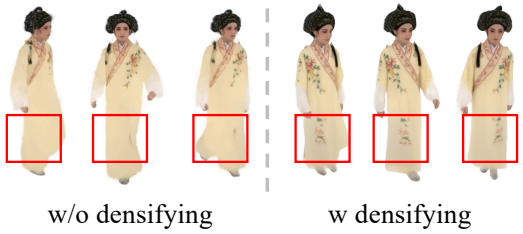
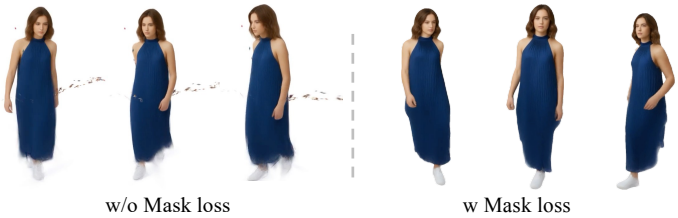


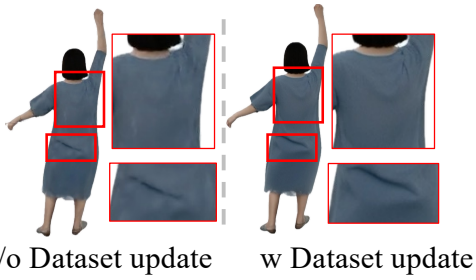

Figure 10. **Sensitivity analysis of the initial noise strength $t_0$.** We visualize restoration results across varying noise strengths. Low noise levels ($t_0 = 0.2, 0.4$) fail to deviate sufficiently from the source, leaving artifacts from the coarse mesh rendering intact. Higher noise levels ($t_0 = 0.6, 0.8$) effectively hallucinate plausible details and correct non-rigid dynamics. Notably, thanks to our self-guidance mechanism, the identity is preserved even at high noise strengths ($t_0 = 0.8$), overcoming the traditional quality-fidelity trade-off.




(a) **Effect of Densification.**




(b) **Effect of Mask Loss.**




(c) **Effect of Dataset Update.**

Figure 11. **Ablations on optimization strategies.** (a) Adaptive densification is crucial for capturing high-frequency texture dynamics. (b) Mask loss regularization is essential to constrain the geometry. (c) Dataset update mitigates over-smoothing caused by inconsistent supervision, allowing the model to converge on sharp, clear details.

the dataset update mechanism allows the optimization to reject inconsistent noise and converge towards high-fidelity results, significantly sharpening fine-grained features such as dress wrinkles.

## F. Results (Extended)

### F.1. Training Efficiency

From a single image, we adopt LHM to obtain the canonical 3D Gaussians, and generate the basic mesh-rigged animations with prepared SMPLX mesh sequences within 1 minute. During re-rendering and 4D optimization, we have $30k$ optimization iterations in total, and update our generated pseudo-ground truth per-$5k$ iterations. Each video re-rerendering (sampling) step takes about 67s in average, and we simultaneously update each trajectory. The overall time cost is about 19 mins. In contrast, PERSONA needs more than 6 hours to create an animation (more than 4 hours for complex data preprocessing and long-sequence video generation, and additional $>$ 1 hour optimization).


Figure 12. Comparison with image-based animation.

## F.2. Discussion with Image-based Animation methods.

To further evaluate the effectiveness of our framework, we compare our method with state-of-the-art image-driven animation models, including Champ and Uni3C, as well as our backbone, Personalized Diffusion. As summarized in Table 5, while image-based methods such as Uni3C achieve competitive rendering quality in terms of Frechet Video Distance, they struggle to maintain high identity consistency, particularly in challenging side-view perspectives. This is reflected in their lower CLIP-Identity scores compared to our approach. Our method consistently outperforms these baselines by leveraging the robust identity priors of the personalized video diffusion model.

Table 5. Quantitative comparison with state-of-the-art image-driven animation methods. Our method achieves a superior balance between motion fidelity and identity preservation.

| Metrics | Champ | Uni3C | Personalized Diffusion | Ours |
|---|---|---|---|---|
| FID ↓ | 196.3 | 132.3 | 138.5 | **112.8** |
| FVD ↓ | 467.2 | **284.4** | 330.6 | <u>289.0</u> |
| CLIP-Identity ↑ | 0.7633 | 0.8357 | 0.8001 | **0.8844** |

A key advantage of our framework over video-based animation methods is the ability to **distill** the pose-controlled video diffusion model into a 4D Gaussian Splatting (4DGS) representation. Traditional video diffusion models require a time-consuming iterative denoising process to generate each sequence. In contrast, once our distillation process is complete, the ***resulting 4D Gaussian representation allows for high-fidelity, real-time rendering of the personalized character in any viewpoint*** . This shift from generative inference to rasterization-based rendering significantly reduces the computational latency, making our approach highly suitable for interactive applications that require both personalized identity and responsive motion control.

## F.3. Results in ActorsHQ dataset [21]

To further assess the generalization capability of our framework, we evaluate ANI3DHUMAN on the high-fidelity ActorsHQ dataset [21]. As shown in Fig. 15 and Fig. 16, our method successfully reconstructs and animates the subject using only a single-view image as input. The results demonstrate that our approach effectively handles challenging articulation scenarios, such as high leg raises, while generating plausible non-rigid dynamics for loose clothing (e.g., skirts). We note that some systematic spatial misalignment between our rendering and the ground truth is observed; this is attributable to inaccuracies in the underlying SMPL parameters estimated from the raw video data, rather than a limitation of the generation pipeline itself. Despite this, the method maintains strong identity preservation and temporal consistency.

## F.4. Additional Qualitative Results

In Fig. 17, we present renderings using dynamic 360-degree camera trajectories across various subjects and complex motions. These results demonstrate that our framework generalizes effectively to diverse identities and actions, maintaining high visual fidelity and temporal consistency from all viewing angles.

## F.5. Limitations

Although our framework achieves high-quality results in 3D human animation, it is subject to the inherent limitations of the underlying representation. Specifically, as we rely on 4D Gaussian Splatting (4DGS), the reconstruction is not strictly lossless. While the method achieves high quantitative metrics (PSNR $\approx$ 35 dB), the discrete nature of the primitives may still result in minor smoothing of extremely high-frequency texture details compared to the source video.


Figure 13. **Visual comparison of sampling strategies.** Cases I: Two girls are walking (Row2/4) and running (Row1/3). The Mesh-Rigged Animation (Input) exhibits unrealistic artifacts, such as unnatural cloth dynamics and blurry edges. Direct Generation suffers from severe identity shift, introducing hallucinations like a bag (Row 2) or a watch (Row 3). ODE Sampling (represented by MCS [74]) fails to recover high-frequency details, leaving garment edges blurry due to the OOD nature of the input. In contrast, Ours successfully restores high-fidelity details and realistic motion while maintaining strict identity consistency.

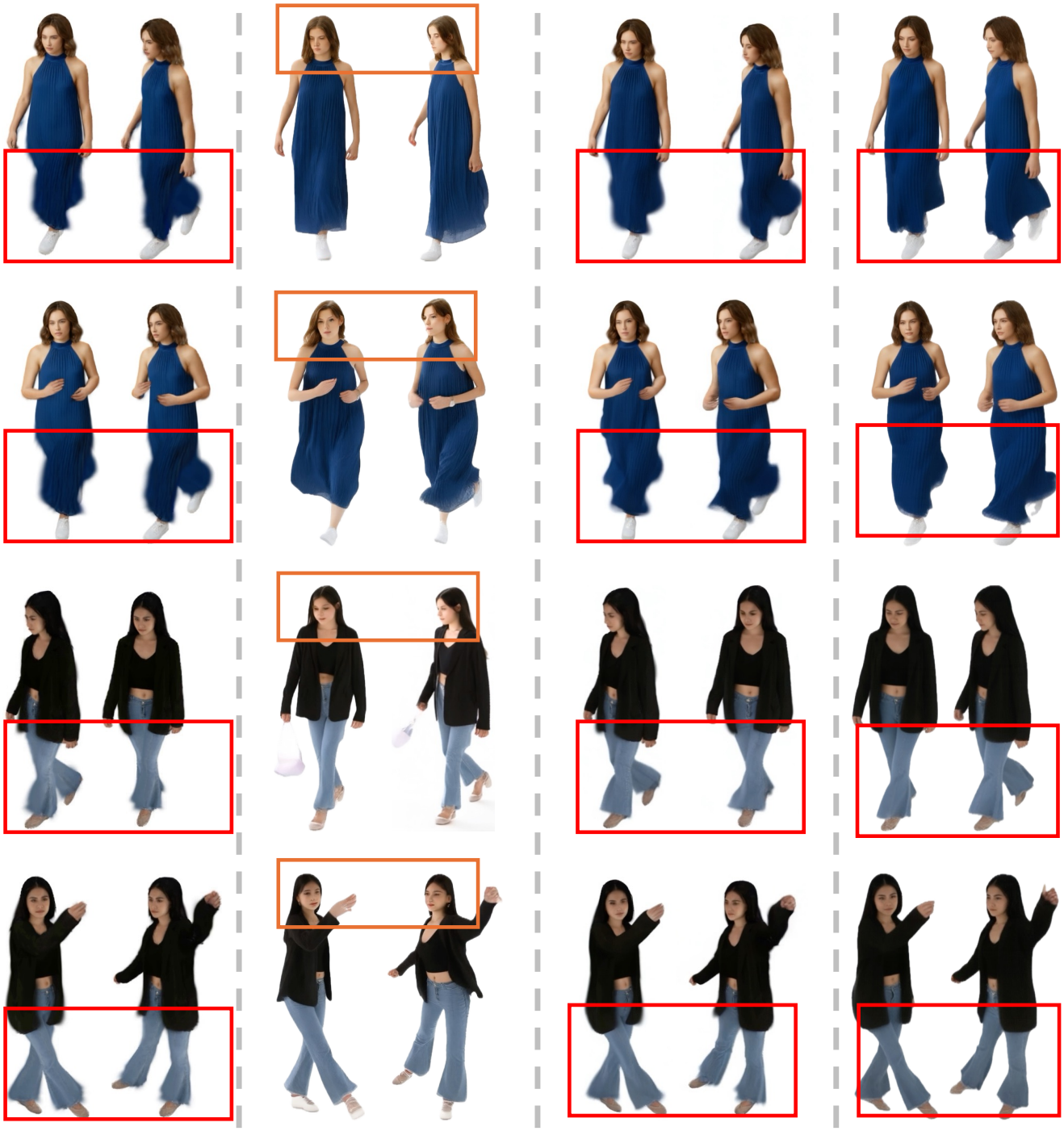



Figure 14. Visual comparison of sampling strategies. Case II: Two girls are walking (Row1/3), running (Row2), dancing (Row4).


Figure 15. **Qualitative evaluation on the ActorsHQ [21] dataset (I).** We show single person with difference motions. The asterisk (*) denotes renderings at a specific viewpoint (elevation $10^{\circ}$, azimuth $0^{\circ}$). Note that slight spatial misalignments between the generation and ground truth are due to inherent errors in the SMPL estimation derived from the source video. Despite relying on a single-view input, our method faithfully preserves human identity and captures complex non-rigid deformations (e.g., dress dynamics), even during extreme poses such as high leg raises.

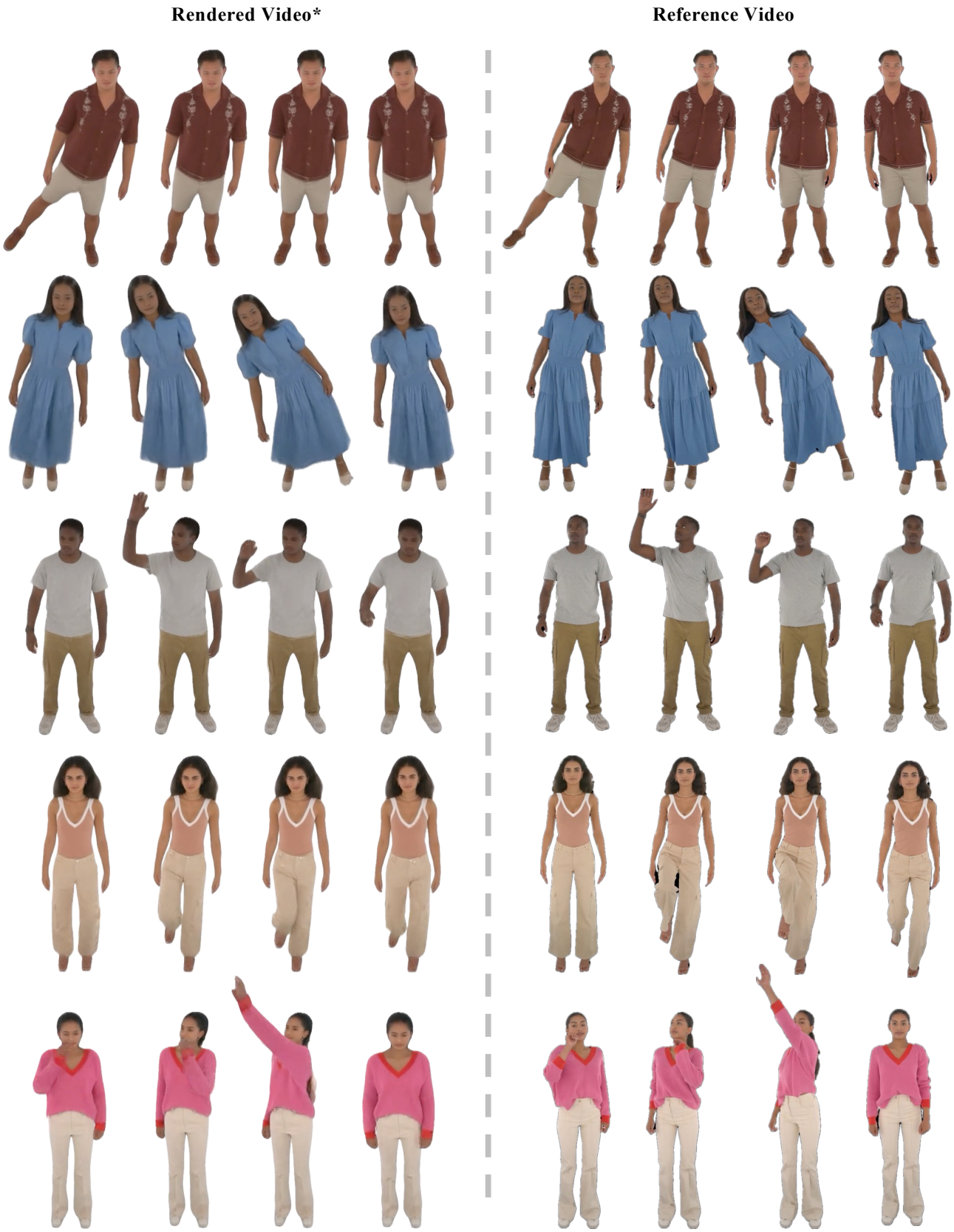



Figure 16. **Human reconstruction results in ActorsHQ [21] dataset (II).** We show different person with diverse motions.

Figure 17. **Additional human animation results.** We visualize diverse subjects performing various motions, rendered with dynamic 360-degree camera trajectories.