Appendix

In this supplementary material, Sec. A presents more qualitative results of DreamWaltz on avatar creation, animation and interaction. Sec. B gives more analysis on the design choices of our method. Section C provides more comprehensive implementation details.

A More Qualitative Results

We provide more results of our proposed method, including more generated avatars in Sec. A.1, more animated avatar sequences in Sec. A.2, and more demonstrations on diverse interactions in Sec. A.3.

A.1 Avatar Creation

We provide more text-to-3D avatar generations in Fig. 12 with a wide range of text prompts including celebrities, popular cartoon/movie characters and text descriptions. Note that DreamWaltz is capable of generating diverse avatars. For instance, we can produce avatars with a human-realistic appearance like "Tiger Woods", avatars wearing intricate clothing such as "Napoleon" and "Marie Antoinette", and avatars tailored to user-provided characteristics like "Blue fairy with wings", among others.

A.2 Avatar Animation

We provide more animation results on six characters as shown in Fig. 13. Please refer to the project page at https://dreamwaltz3d.github.io/ for more animation sequences.

A.3 Diverse Interaction

We provide more results of diverse interactions in Fig. 14, including: avatar-object, avatar-scene, and avatar-avatar interactions. Please refer to the videos on project page at https://dreamwaltz3d.github.io/ for the full sequences.

B More Analysis

B.1 Visualization of SDS Gradients

We visualize the SDS supervision gradients $\|\boldsymbol{\epsilon}_\phi(\mathbf{z}_t; y, t) - \boldsymbol{\epsilon}\|$ for NeRF renderings in Fig. 15 (a) and the denoised images derived from noise predictions $\boldsymbol{\epsilon}_\phi(\mathbf{z}_t; y, t)$ in Fig. 15 (b). These visualizations are based on the text prompt y of "superman" and t of 980, while additional conditioning of depth and skeleton is provided in the second and third row, respectively, in accordance with the current rendering viewpoint of NeRF. It is evident that depth and skeleton images offer more informative optimization gradients compared to text alone. However, depth images heavily rely on the SMPL prior, leading to gradients that conform tightly to the avatar's skin, resulting in the disappearance of superman's cape. On the other hand, skeleton images as adopted by DreamWaltz provide both informative and flexible supervision, accurately capturing the avatar's shape, pose, and intricate details such as the cape.

B.2 Effects of Random-Pose Optimization on Avatar Quality

In Fig. 16, we present visualizations of the avatars obtained at various stages, all depicted in a canonical pose. In Stage I, a static avatar is generated by optimizing its 3D representation with the canonical pose. In Stage II, the system undergoes training on randomly sampled human poses to facilitate animation learning. Although not our primary objective, this design enables the generated avatar to further refine its appearance with different poses. As a result, minor adjustments are observed, typically leading to sharper details. However, in some cases, the geometry may become more simplified, aligning more closely with the SMPL prior.

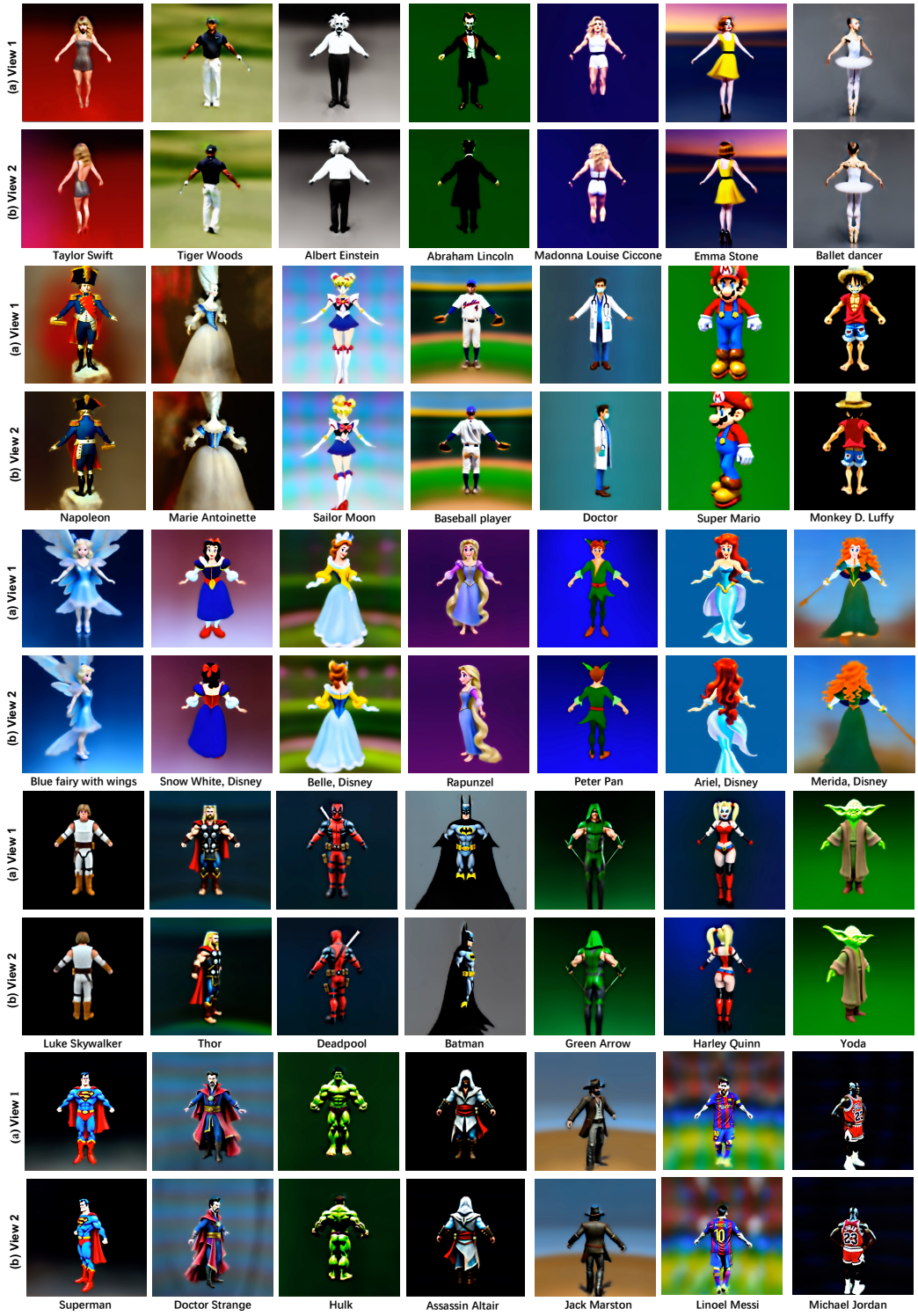


Figure 12: Text-to-3D avatars generated with DreamWaltz, each displayed for two views.

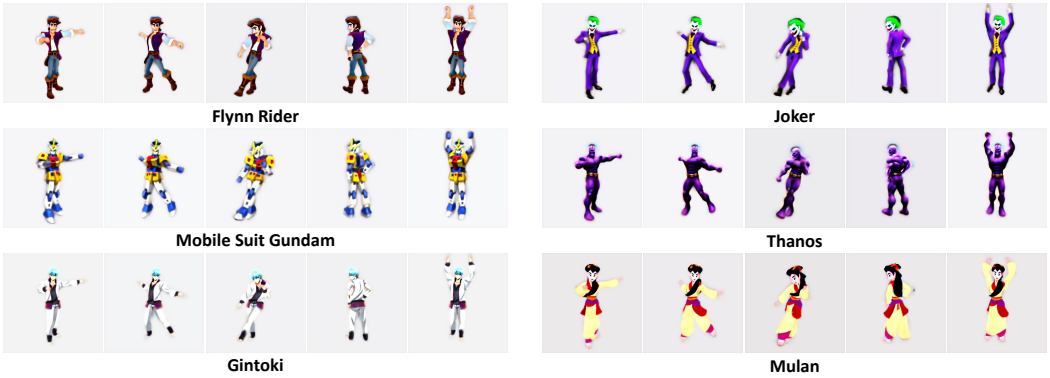
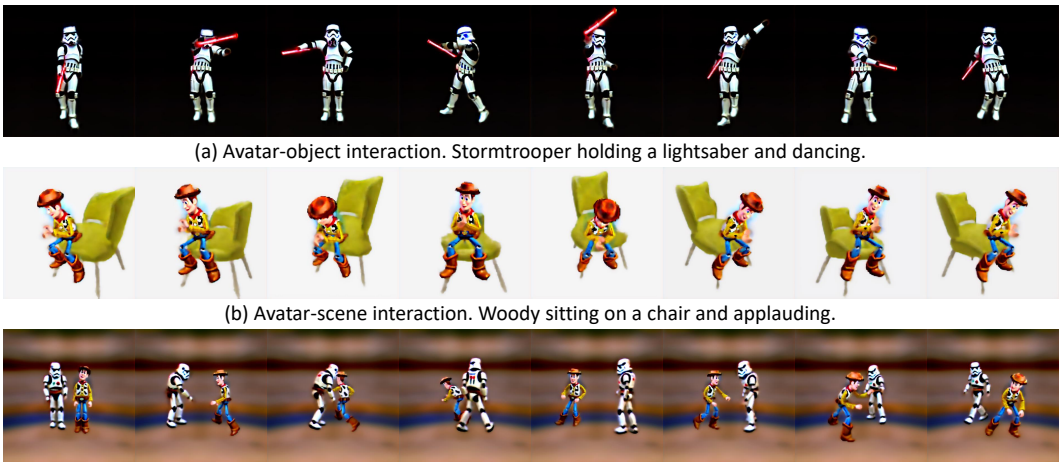


Figure 13: Avatar animations on six characters.

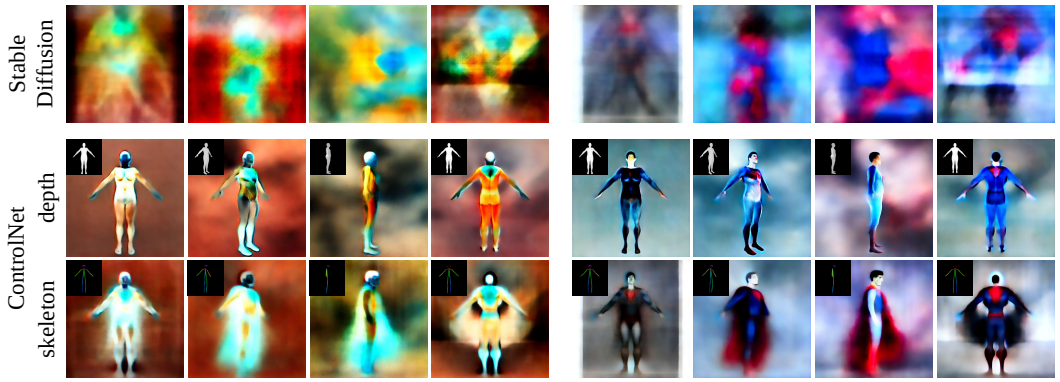


(a) Avatar-object interaction. Stormtrooper holding a lightsaber and dancing.

(b) Avatar-scene interaction. Woody sitting on a chair and applauding.

(c) Avatar-avatar interaction. Woody and Stormtrooper playing basketball.

Figure 14: Avatar animations with diverse interactions.



(a) SDS gradients (b) One-step denoised images

Figure 15: Visualization of the SDS gradients (a) and the corresponding denoised images (b), given the text prompt "superman". The second and third rows are conditioned on additional depth and skeleton images, respectively, as indicated in the upper left corner of each visualization. It is clear that the skeleton image as adopted by DreamWaltz provides more informative supervision compared to text alone. Skeleton conditioning is also less restrictive than depth conditioning, successfully avoiding the disappearance of superman's cape.

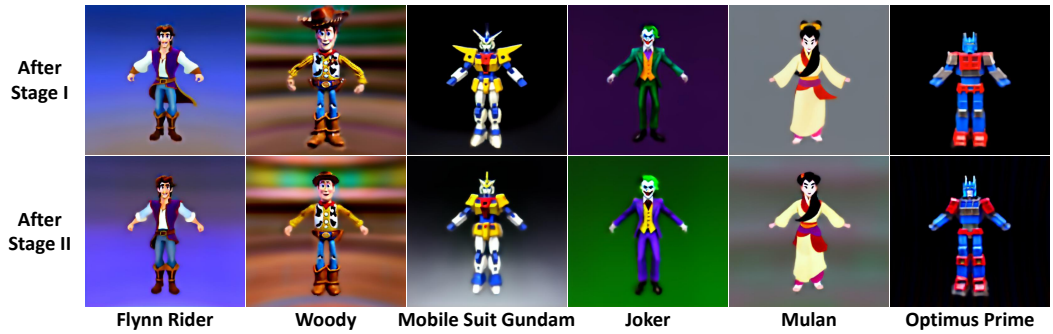


Figure 16: Visualization of canonical avatars obtained at different stages. The optimization at Stage II slightly changes the shape and appearance of avatars, resulting in sharper details (e.g., Woody's hat) but sometimes more simplified geometry (e.g., Flynn's clothes).

B.3 Single-stage Training vs. Two-stage Training

In Fig. 17, we present a qualitative comparison between single-stage training (Stage II only) and two-stage training (Stage I + Stage II) approaches using our proposed framework, specifically for an avatar animation (e.g., on a dance motion sequence). When applied to different characters, the single-stage strategy may result in problematic body topology and noticeable artifacts. In contrast, the two-stage strategy effectively mitigates these issues, leading to improved visual quality. When employing a single-stage strategy with end-to-end training, the model is required to simultaneously learn the generation of avatar geometry and appearance, along with animation. This introduces complex optimization dynamics, leading to potential slower optimization and sub-optimal results.

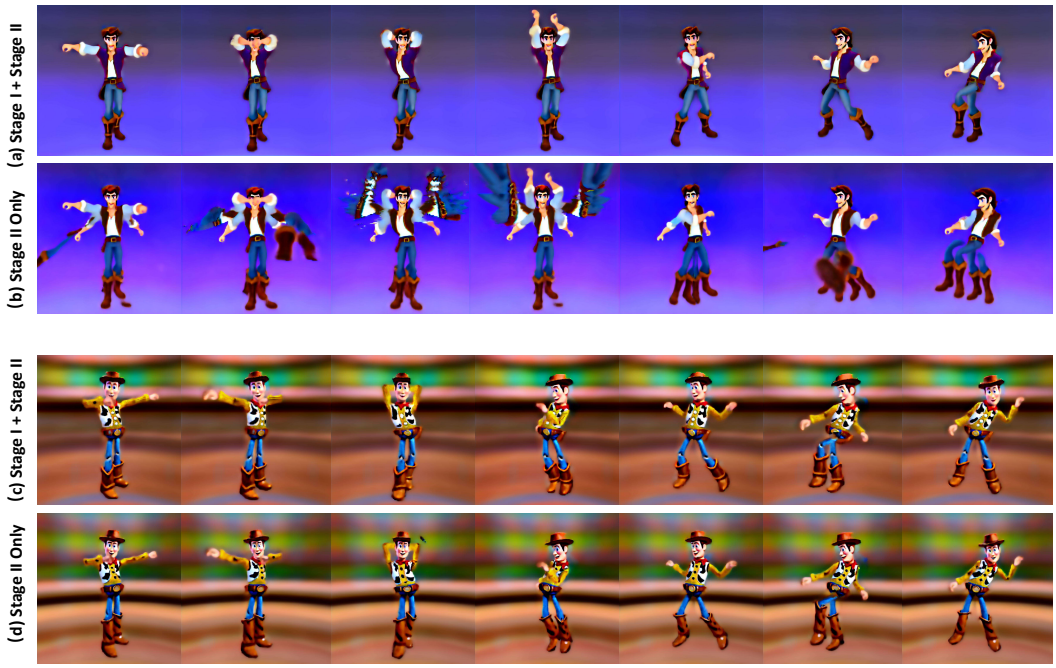


Figure 17: Qualitative comparisons of single-stage training (Stage II only) with two-stage training (Stage I + Stage II) on a dance motion sequence, both based on our proposed framework. For different characters, the single-stage strategy may suffer incorrect body topology and severe artifacts, as shown in (b). In contrast, the two-stage strategy can relieve these issues (from (b) to (a)) with better visual quality (from (d) to (c)). The one-stage strategy tends to be subjected to slow optimization speed and sub-optimal results.

B.4 Effects of Joint Optimization for Scene Composition

Benefiting from DreamWaltz, we can create diverse animatable avatars that are prepared to engage in scenes with interactions. One approach would be to simply render different animatable avatars together in a scene. However, such composition is susceptible to issues such as artifacts and unnatural interactions. To further improve the scene quality, we could fine-tune for each specific scene. As shown in Fig. 18, fine-tuning brings noticeable improvements, such as enhancements to Woody's hat and boots, as well as more realistic "hands bumping" interactions.

Text Prompt for Scene Refinement: "***Woody in Toy Story playing basketball with Stormtrooper***"

Figure 18: To further enhance the visual quality of complex scene generation involving multiple avatars and interactions, scene refinement (i.e. fine-tuning with our proposed 3D-consistent SDS) can be applied to eliminate artifacts. As depicted in the frames marked with blue boxes, fine-tuning brings noticeable improvements, such as enhancing the appearance of Woody's boots and achieving more realistic "hands bumping" effects.

B.5 Realistic Animation with Pose-dependent Changes

To further demonstrate our animation performance, we provide animation results of complex character "Elsa" (with long hair and skirt) using our animation method, in comparison with animating extracted mesh with the commercial application *Mixamo*. As demonstrated in Fig. 19, with our method, Elsa's skirt and hair exhibit significantly more natural displacements and movements (as highlighted in yellow and red, respectively) as pose changes.

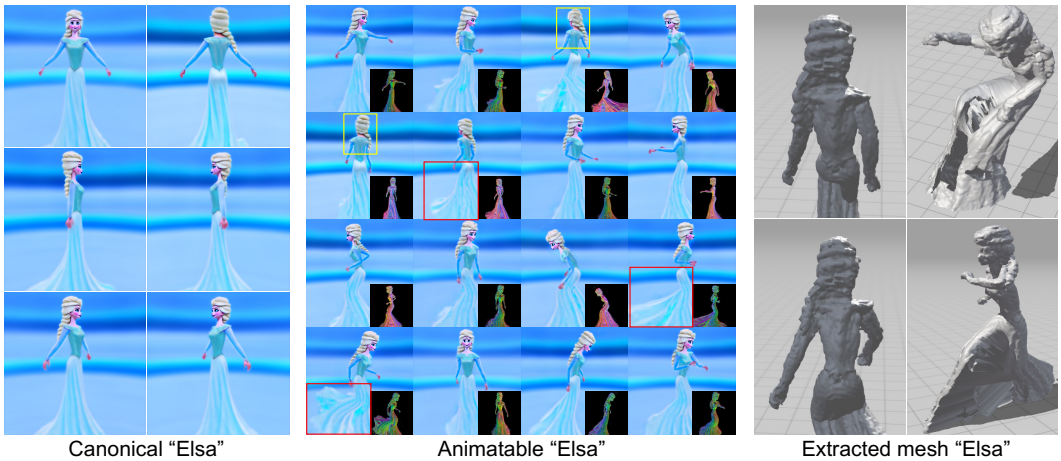


Figure 19: Animation results of complex avatar (Elsa with long hair and skirt). Our animation method could achieve high-quality realistic animation with pose-dependent changes (hair displacement and skirt movements as highlighted in yellow and red respectively), while animation with rigged mesh fails with hair stuck to left arm and unrealistic skirt.

C More Implementation Details

Diffusion Guidance. We adopt ControlNet [48] with Stable-Diffusion v1.5 [34] as the backbone to provide 2D supervision. Specifically, we utilize the score distillation sampling (SDS) technique introduced by DreamFusion [31] to obtain the back-propagation gradients of 3D avatar representation. During training, we randomly sample the timestep from a uniform distribution of $[20, 980]$, and the classifier-free guidance scale is set to 50.0. The weight term $w(t)$ of SDS loss is set to 1.0, and we normalize the SDS gradients to stabilize the optimization process. The conditioning scale for ControlNet is set to 1.0 by default.

The proposed DreamWaltz utilizes two types of conditioning: text prompt and skeleton image. The text prompt is given by the user to provide avatar description. View-dependent text augmentation from DreamFusion [31] is also used:

$$\begin{cases} \text{“front view of...”} & \theta_{\text{cam}} \in [0^\circ, 90^\circ] \\ \text{“backside view of...”} & \theta_{\text{cam}} \in [180^\circ, 270^\circ] \\ \text{“side view of...”} & \text{otherwise,} \end{cases}$$

where θ_{cam} denotes the azimuthal angle of camera position. The skeleton image is exported from the 3D SMPL mesh, where the rendering view is required to be consistent with the rendering view of NeRF for training.

NeRF Rendering. We adopt Instant-NGP [25] as the implicit avatar representation. The ray marching acceleration based on occupancy grid is disabled for dynamic scene rendering. The 3D avatar representation renders “latent images” in the latent space of $\mathbb{R}^{64\times64\times4}$ following Latent-NeRF [21], where the “latent images” can be decoded into RGB images of $\mathbb{R}^{512\times512\times3}$ by the VAE decoder of Stable Diffusion [34]. During training, the camera positions are randomly sampled in spherical coordinates, where the radius, azimuthal angle, and polar angle of camera position are sampled from $[1.0, 2.0]$, $[0, 360]$ and $[60, 120]$, respectively.

Optimization. Throughout the entire training process, we use Adam [16] optimizer with a learning rate of 1e-3, and batch size is set to 1. For the canonical avatar creation stage, we train the avatar representation for 30,000 iterations, which takes about an hour on a single NVIDIA 3090 GPU. For the animatable avatar learning stage, the avatar representation and the introduced density weighting network are further trained for 50,000 iterations. Inference takes less than 3 seconds per rendering frame. We further fine-tune the hybrid avatar representations for 30,000 more steps for scenarios with multiple avatars and complex interactions.

Dataset. To create animation demonstrations, we utilize SMPL-format motion sequences from the 3DPW [42] and AIST++ [17] datasets to animate avatars. SMPL-format motion sequences extracted from in-the-wild videos are also used.