Arc2Avatar: Generating Expressive 3D Avatars from a Single Image via ID Guidance
Supplementary Material

Dimitrios Gerogiannis Foivos Paraperas Papantoniou Rolandos Alexandros Potamias
Alexandros Lattas Stefanos Zafeiriou

Imperial College London, UK

{d.gerogiannis22, f.paraperas, r.potamias, a.lattas, s.zafeiriou}@imperial.ac.uk

This document offers supplementary details about our method that could not be included in the main paper due to space constraints.

1. Optimization of the Facial Area

In this section, we illustrate the avatar optimization process for the facial area, highlighting the efficiency of our framework in swiftly capturing the person's facial features through strategic initialization, before advancing to the optimization of the entire head.

1.1. Fitting Splats to the Mean Facial Surface

As discussed in the main paper (Sec. 3.4), it is beneficial to initialize person-specific avatar generation using a set of Gaussian Splats representing the mean colored facial surface, rather than just the upsampled FLAME point cloud. This approach requires an initial optimization of all splats' parameters (including covariance) based on mesh renderings with the mean texture, as merely assigning the RGB values of each vertex to the point cloud would still result in a discontinuous representation. This fitting process is described in Sec. 3.4 and is further illustrated in Fig. 1 for clarity. Please note that this step is independent of the input subject and is performed only once. The fitted splats, being identical for all subjects, serve as a precomputed initialization for subsequent subject-specific optimizations.

1.2. Personalization of the Facial Area

As described in the paper, our person-specific SDS optimization begins with the mean texture-fitted splat and proceeds in two phases: first optimizing the facial region, followed by optimizing the entire head. The initial face-only phase consists of 500 iterations, during which we sample views with azimuth angles in the range $[-110, 110]$ and elevation angles in the range $[60, 90]$. To capture finer details, we zoom in during this phase, using a field of view of 0.4. In

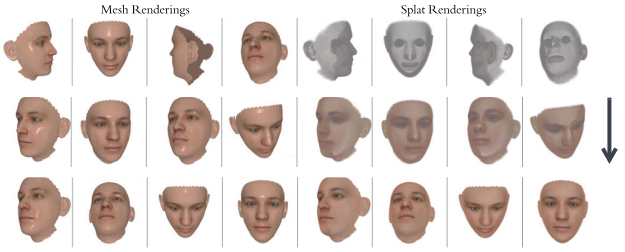

Figure 1. **Initial mean texture fitting.** The splats in the facial area are optimized based on mesh renderings with the mean texture. In the end, the splats closely replicate the mean textured mesh.

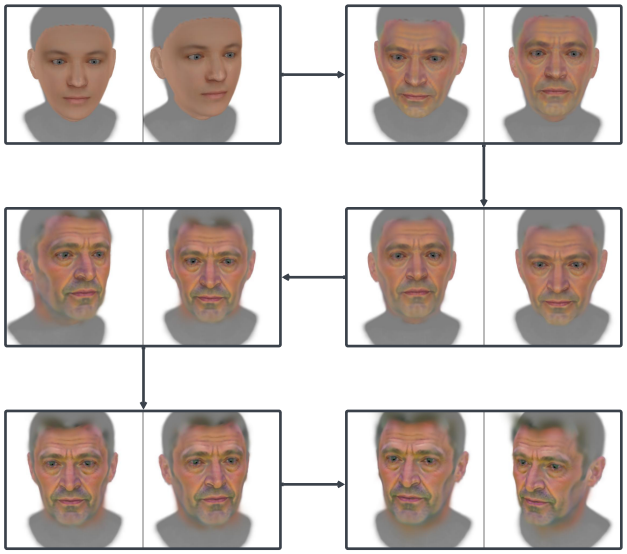

Figure 2. **Face-only optimization progression.** The splats in the facial area smoothly transition from the mean texture to the input subject's face.

Fig. 2, we illustrate the progression of this initial optimiza-

tion, showing how the mean texture-fitted splat smoothly transforms into the subject's face while maintaining correspondence with the underlying template mesh, highlighting the effectiveness of our approach.

2. Ablation Studies

Below, we provide additional results demonstrating the necessity of various components in our pipeline.

2.1. Importance of Mean Texture Initialization

We argue that template regularizers alone are insufficient to guarantee template correspondence in an SDS setup. To support this claim, we present results from the initial facial area optimization phase, comparing scenarios with and without mean texture initialization, as shown in Fig. 3.

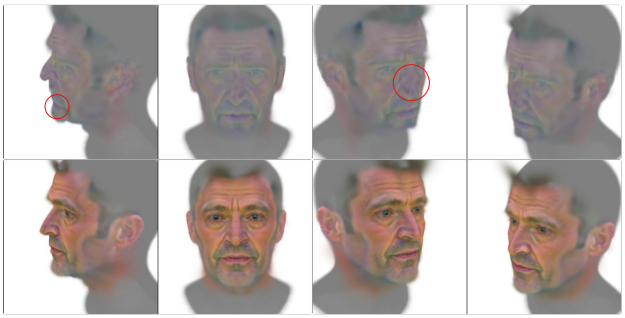

Figure 3. **Mean texture initialization impact.** Although template regularization achieves geometrical correspondence, the absence of the proposed initialization (top) leads to significant texture misalignment, disrupting the overall correspondence. Without mean texture initialization, facial features are often placed to incorrect locations (e.g., the mouth placed on the chin) or exhibit artifacts, such as duplicate features (e.g., the nose). In contrast, mean texture initialization (bottom) ensures proper correspondence from the early stages of the process.

2.2. Arc2Face Augmentation and View Embeddings

Furthermore, we demonstrate the necessity of both LoRA-based Arc2Face fine-tuning and the use of view text embeddings in conjunction with the identity embedding for conditioning. These processes are crucial for achieving realistic 3D avatars without Janus artifacts or inconsistencies.

As described in the main paper, we create view-enriched embeddings by blending the default identity embedding $\mathbf{c}_{\text{default}}$ with the view embedding $\mathbf{c}_{\text{view}}$ using the formula:

$$\mathbf{c}_d = b \cdot \mathbf{c}_{\text{default}} + (1 - b) \cdot \mathbf{c}_{\text{view}}, \tag{1}$$

where $b \in [0, 1]$ balances the influence of identity and view.

In Fig. 4, we present renderings of the avatar produced after the first half of the optimization steps for five different variations of our method:

1. **Default Arc2Face Model:** Using the default ID-conditioned Arc2Face model as the guidance model without any modifications.
2. **LoRA-Extended Model without View Embeddings:** Using the LoRA-extended model but without view embeddings, effectively setting the view embedding weight to zero ($1 - b = 0$).
3. **Strong View Embedding Weight** ($1-b = 0.45$)**:** Using a strong weight for the view embedding to emphasize view information.
4. **Medium View Embedding Weight** ($1 - b = 0.3$)**:** Using a medium weight for the view embedding, providing a balanced influence between identity and view.
5. **Our Method** ($1 - b = 0.15$)**:** Using the blending factor we chose for our method, which we found to offer the best trade-off between identity preservation and view consistency.

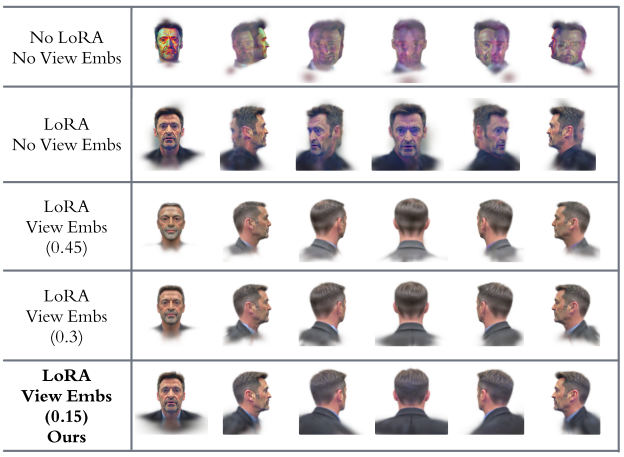

Figure 4. **Impact of augmenting Arc2Face for 360° generation and using view embeddings during distillation.** As expected, the default model is limited to modeling the frontal view, resulting in multiple inconsistencies and Janus artifacts in other views, as well as oversaturated colors, rendering it unsuitable for guidance in its default state. The LoRA-extended model without view embeddings performs better, achieving good identity preservation and improved side views, but it still exhibits Janus effects in the back view. Using strong weights for view embeddings generates very consistent heads with good back and side views but significantly reduces identity fidelity. In contrast, our selection of a low weight for view embeddings (final row) achieves the best of both worlds, combining identity preservation with consistency in the generated heads, and eliminating Janus effects.

3. Additional Qualitative Results

In this section, we showcase additional 3D avatars generated by our method for subjects with significantly diverse characteristics. As can be seen in Fig. 5, our method exhibits strong generalizability, capable of producing high-

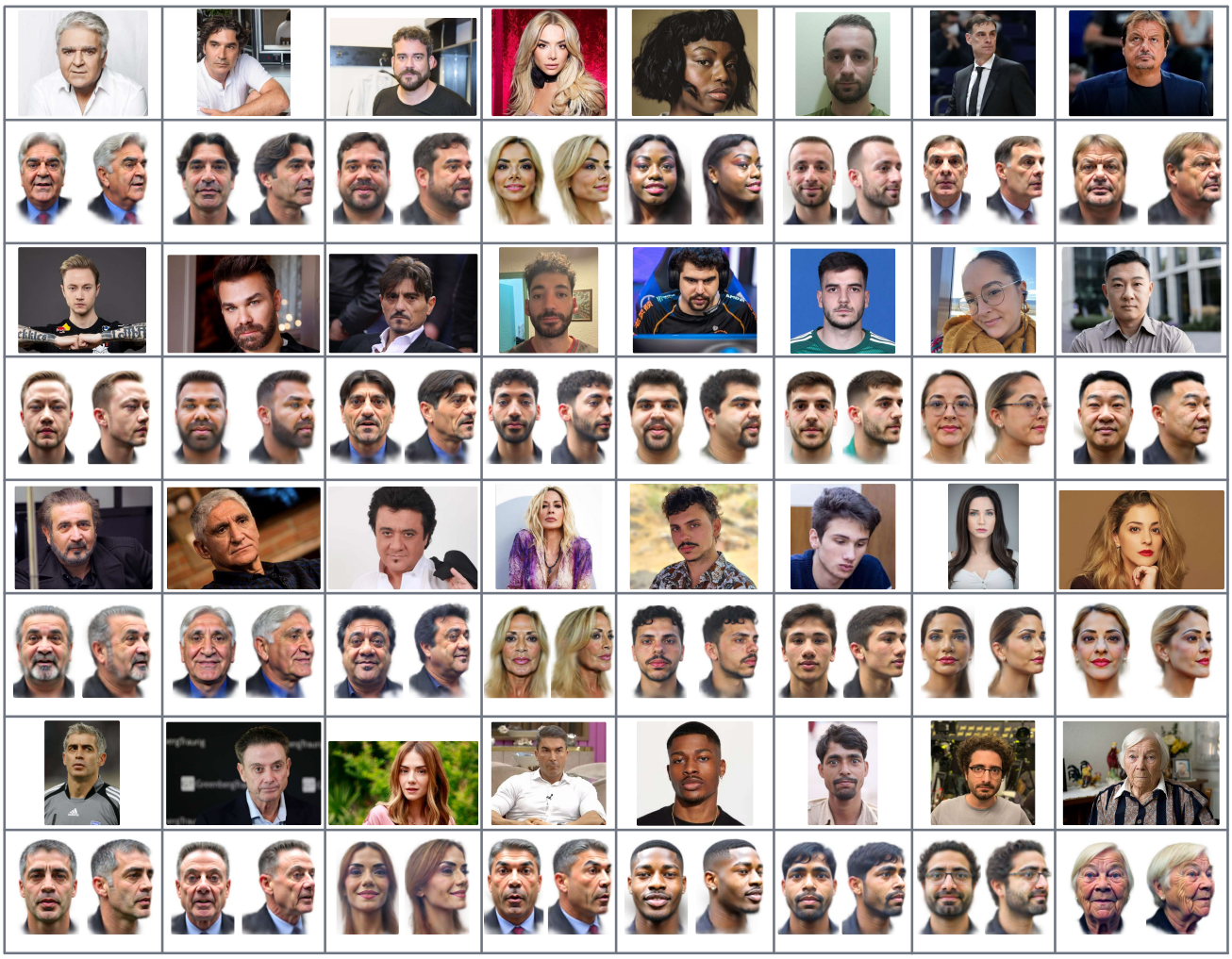

Figure 5. **Arc2Avatar is not limited to celebrities.** Our method exhibits strong generalization, providing realistic and consistent 3D avatars for individuals of different ages, ethnicity, and backgrounds.

fidelity, ID-consistent 3D heads for any individual.

Moreover, in Fig. 6, we provide renderings from multiple perspectives for many samples, demonstrating our method's 3D consistency and fidelity. Notably, our approach effectively generates realistic views, including challenging backhead perspectives, which are inferred solely from frontal input images thanks to our careful adaptation of Arc2Face for diverse view generation using frontal inputs.

4. Additional Qualitative Comparisons

Although not directly comparable to our method, we qualitatively compare with a 3D-aware portrait synthesis method (GRAM-HD) [7] and a separate multi-view diffusion approach (Wonder3D) [4] in Fig. 7.

5. Failure Cases

As discussed in the main paper, our method has certain limitations, including the introduction of artifacts and the occasional addition of expressions by Arc2Face in the neutral optimization stage despite our efforts to enforce consistency with the neutral mesh, disrupting correspondence with the template. In Fig. 8, we present examples showing these issues.

6. Implementation Details

6.1. Arc2Face Fine-Tuning

We fine-tuned the LoRA-augmented Arc2Face model following a setting similar to [6]. In particular, we used a resolution of 512×512 pixels for our synthetic 360° dataset and trained the model with AdamW [5] and a learning rate

Figure 6. **Renderings of generated 3D avatars from diverse viewpoints.** Our method extends beyond realistic frontal views to produce complete 3D head models that can be rendered from any angle.

of 1e-4 for the LoRA layers, using one NVIDIA A100 GPU and a batch size of 4. We trained for 100K iterations, as further fine-tuning caused noticeable identity loss, making it harder for SDS to handle these inconsistencies.

Figure 7. **Qualitative comparisons.** Competing methods generate plausible frontal and slightly side views, but struggle with side and back views, demonstrating the advantages of our method.

6.2. FLAME-based Point Cloud Initialization

We initialize the splats based on the FLAME mesh, which consists of $N_{\text{original}} = 5023$ vertices. However, given the low vertex count, we first perform dense sampling of the mesh. Maintaining consistency in the upsampling process is essential to ensure that when expression blendshapes are applied to the facial region, the resulting deformations are consistently upsampled and accurately incorporated into the upsampled facial mesh. To achieve this, we apply the `subdivide()` method from the `trimesh` [1] library, which implements the Midpoint Subdivision algorithm. This process upscales the original mesh to $N_{\text{upsampled}} = 79936$ vertices, with the majority concentrated in the facial area of interest ($N_{\text{face}} = 70033$ vertices) and the remaining $N_{\text{head}} = 9903$ vertices allocated to the rest of the head. Since we are only concerned with maintaining consistent upsampling within the facial region, we separate the mesh into facial and head components. The head component is then independently upsampled to $N_{\text{head}} = 73050$ vertices. Finally, the facial and head meshes are reconnected, resulting in a unified point cloud that serves as the initialization for the optimized splat $G_{\text{init}}(\mathbf{x})$.

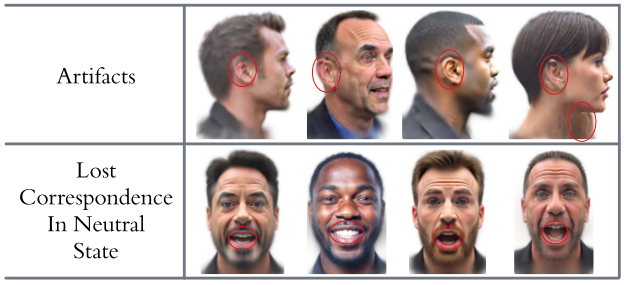

Figure 8. **Failure cases.** Artifacts may appear around the ears and neck regions. Additionally, certain inputs can bias the optimization towards smiling or surprised expressions, despite the underlying neutral mesh. Nevertheless, the avatars consistently preserve the individuals' identities.

6.3. SDS-ISM Parameters

Our distillation framework is based on ISM [3]. The settings detailed below are presented in full correspondence with their method, following the same format.

6.3.1. Guidance Parameters

As discussed in the main paper, our strong prior and carefully designed task-specific SDS process, along with settings refined through experimentation, eliminate the need for the high guidance scale typically associated with SDS approaches, effectively avoiding color issues. Specifically, we employ a scale of 1 and the Perp-Neg algorithm [2], and, following the notation of [3], we use $\delta_T = 40$ paired with $\delta_S = 20$, utilizing 20 inversion steps. This results in 3D avatars that exhibit high detail and natural color.

6.3.2. Camera Parameters

We utilize a random camera sampling strategy with progressively relaxed view ranges during training. The initial camera configurations are:

- Radius range: $[5.2, 5.5]$.
- Maximum radius range: $[4.2, 5.2]$.
- Field of view (FoV) range: $[0.53, 0.53]$.
- Maximum FoV Range: $[0.3, 0.7]$.
- Elevation angle range (θ): $[40°, 100°]$.
- Azimuth angle range (ϕ): $[-180°, 180°]$.

Starting from iteration 2000, we progressively relax the camera view ranges every 2000 iterations by scaling the parameters:

- FoV factor: $[0.8, 1.1]$.
- Radius factor: 0.95.

6.3.3. Optimization Parameters

We train our avatars with a rendering resolution of 512×512 pixels for 6000 iterations on a single NVIDIA RTX 4090 GPU (24GB) using a batch size equal to 4. Optimizing an avatar for an input subject takes approximately 80 minutes, and the final avatar typically consists of nearly 110K Gaussians.

The optimization is performed using the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1\text{e}{-15}$. The learning

rates for different parameters are scheduled to decay exponentially from their initial values to final values over the course of training, using a delay multiplier of 0.01:

- Position (μ): $\text{lr}_{\text{init}} = 1.6\text{e}{-4}$, $\text{lr}_{\text{final}} = 1.6\text{e}{-6}$.
- Color ($\mathbf{f}$): $\text{lr}_{\text{init}} = 5\text{e}{-3}$, $\text{lr}_{\text{final}} = 3\text{e}{-3}$.
- Opacity (α): $\text{lr} = 5\text{e}{-2}$.
- Scaling ($\mathbf{s}$): $\text{lr}_{\text{init}} = 5\text{e}{-3}$, $\text{lr}_{\text{final}} = 1\text{e}{-3}$.
- Rotation ($\mathbf{r}$): $\text{lr}_{\text{init}} = 1\text{e}{-3}$, $\text{lr}_{\text{final}} = 2\text{e}{-4}$.

6.4. Splat Modification Strategy

To refine the splats in the non-facial areas, we initiate densification and pruning at iteration 1000, performing them every 500 iterations until 5000. During this period, opacity resets are also applied every 1000 iterations. In the final 1000 iterations, we further refine the splats by pruning disconnected splats every 100 iterations to remove isolated noise and applying targeted pruning based on opacity and size every 200 iterations.

6.5. Camera Sampling Strategy

Given the approximate symmetry of human heads, we observed that sampling an equal number of front and back views during training was more beneficial than randomly sampling any azimuth angle. To achieve this, we enforced the sampling of four azimuth angles for each training step:

- Two angles from the frontal range $[-90°, 90°]$: one from $[-90°, 0°)$ and one from $[0°, 90°)$.
- Two angles from the back range $[-180°, -90°) \cup (90°, 180°]$: one from $[-180°, -90°)$ and one from $(90°, 180°]$.

This strategy ensured a balanced and diverse set of views, encompassing frontal, back, and side perspectives.

6.6. Template Regularization

As discussed in the main paper, we employ strong template proximity regularizers, specifically the L_2 distance regularizer and the Laplacian difference regularizer. Through experimentation, we found that using high weights for these regularizers leads to very strong template correspondence. Therefore, we selected a value of $1\text{e}{+8}$ for both.

References

[1] Trimesh. https://trimesh.org Version 4.3.1. 5

[2] Mohammadreza Armandpour, Huangjie Zheng, Ali Sadeghian, Amir Sadeghian, and Mingyuan Zhou. Re-imagine the negative prompt algorithm: Transform 2d diffusion into 3d, alleviate janus problem and beyond. *arXiv preprint arXiv:2304.04968*, 2023. 5

[3] Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6526, 2024. 5

[4] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. *arXiv preprint arXiv:2310.15008*, 2023. 3

[5] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 3

[6] Foivos Paraperas Papantoniou, Alexandros Lattas, Stylianos Moschoglou, Jiankang Deng, Bernhard Kainz, and Stefanos Zafeiriou. Arc2face: A foundation model for id-consistent human faces. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 3

[7] Jianfeng Xiang, Jiaolong Yang, Yu Deng, and Xin Tong. Gram-hd: 3d-consistent image generation at high resolution with generative radiance manifolds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2195–2205, 2023. 3