Appendix

A. Overview of Appendices

Our appendices contain the following additional details:

- **Sec. B** provides implementation details of the baseline methods, including configurations and training settings.
- **Sec. C** specifies the evaluation metrics used in our experiments, detailing their definitions and computational methods.
- **Sec. D** describes the training setup, including hyperparameter choices, optimization strategies, and training procedures.
- **Sec. E** provides additional experimental results.
- **Sec. F** presents the model details, including architecture components, parameter settings, and additional design choices.

B. Baseline implementation

B.1. DreamBooth

We employ DreamBooth to achieve personalized face aging, leveraging its ability to fine-tune models for specific individuals while preserving their identity. The entire training process consists of two stages to ensure that the generated faces exhibit both the desired age-related characteristics and the original identity information.

Stage 1: Fine-tuning for Personalized Aging Model. In the first stage, we aim to train the model to generate faces with specific target age characteristics. To achieve this, we collect a dataset of facial images corresponding to the desired age range and use them to fine-tune the model. During training, we employ two types of text prompts:

- **Instance Prompt:** We use the text `"a photo of xx years old man/woman"` (e.g., `"a photo of 60 years old man"`) to explicitly instruct the model to generate facial features that match the target age.
- **Class Prompt:** We utilize a more generic class-level prompt, `"a photo of man/woman"`, to ensure diversity in the generated images and prevent catastrophic forgetting.

Based on these image-text pairs, we fine-tune the model to adapt to the desired age transformation. The optimization process is guided by the following loss function:

$$\begin{aligned}\mathcal{L} =& \mathbb{E}_{x,c,t}\left[w_t \left\|\hat{x}_\theta(\alpha_t x + \sigma_t \epsilon, c) - x\right\|_2^2\right] \\ &+ \lambda \mathbb{E}_{x_{pr},c_{pr},t'}\left[w_{t'} \left\|\hat{x}_\theta(\alpha_{t'} x_{pr} + \sigma_{t'} \epsilon', c_{pr}) - x_{pr}\right\|_2^2\right]\end{aligned} \tag{1}$$

where:

- x: A real image of a person with a specific age and identity.
- c: The **Instance Prompt**, which explicitly specifies the target age. This ensures that the generated image matches the target age.
- x_{pr}: A class-level image sampled from a general dataset, not associated with a specific identity.
- c_{pr}: The **Class Prompt**, a generic description, which helps maintain diversity and prevents catastrophic forgetting.
- $\hat{x}_\theta(\cdot)$: The denoised output computed by the model.
- α_t, σ_t: Scaling factors for the diffusion and denoising process.
- ϵ, ϵ': Gaussian noise sampled from a standard normal distribution.
- $w_t, w_{t'}$: Time-dependent loss weights.
- λ: A hyperparameter balancing the two loss terms.

The first term is the personalization loss, which trains the model to reconstruct the specific identity images. The second term is the class preservation loss, ensuring that the model does not overfit the personalized data and forget general class-level features (preventing catastrophic forgetting).

Stage 2: Personalized Identity Token Fine-tuning. In the second stage, we aim to associate the source face with a specific identity token, enabling precise control over age transformation during inference. To achieve this, we introduce a unique token `"sks"` as an identity mapping and fine-tune the model using the following setup:

- **Instance Prompt:** We use `"a photo of sks man/woman"` as the instance prompt, linking the source face to the identity token `"sks"`.
- **Class Prompt:** Similar to the first stage, we use `"a photo of man/woman"` as the class prompt to ensure that the generated faces retain realistic gender attributes without overfitting to a particular dataset.

Through this second fine-tuning stage, the model learns to generate age-transformed faces while preserving the original identity. The loss function remains similar to the first stage, combining reconstruction loss and class-based constraints to maintain realism and diversity.

LoRA-Based Fine-Tuning for Efficient Adaptation. To improve training efficiency and reduce redundant model parameters, we employ LoRA (Low-Rank Adaptation) in DreamBooth fine-tuning. The updated model weights are represented as:

$$W' = W + \lambda_1 \Delta W_1 + \lambda_2 \Delta W_2 \tag{2}$$

where W is the original pre-trained model weight, ΔW_1 and ΔW_2 are low-rank weight update matrices for different training stages, and λ_1, λ_2 are tunable hyperparameters. By adjusting λ_1 and λ_2, we can flexibly balance the degree of editability and identity preservation. This approach enhances training stability while allowing more precise age transformation control during inference.

During inference, we use "`a photo of xx years old sks man/woman`" to generate target image.

B.2. IP-Adapter-FaceID

IP-Adapter-FaceID is a personalized face generation method based on image-conditioned prompts, enabling precise face aging transformations. By leveraging multiple face embeddings, we can achieve personalized age modifications while preserving identity characteristics.

Computing the Target Age Embedding We first collect a large dataset of face images corresponding to a specific target age group. These images serve as a reference to capture the general age-related facial attributes. To extract a representative embedding for the target age, we compute the average embedding of the collected facial images as follows:

$$\mathbf{embs}^* = \frac{1}{N} \sum_{i=1}^{N} E(I_i) \tag{3}$$

where $E(I_i)$ represents the embedding extracted from each facial image I_i, and N is the total number of images in the dataset. This average embedding serves as a generalized representation of the facial characteristics associated with the target age.

Extracting the Source Face Embedding Next, we extract the embedding of the source face, denoted as:

$$\mathbf{embs}_{\text{src}} = E(I_{\text{src}}) \tag{4}$$

This embedding captures the identity-specific characteristics of the input face before any aging transformation.

Generating the Hybrid Face Embedding To obtain the final embedding representation for face generation, we combine the source face embedding with the target age embedding using a weighted interpolation:

$$\mathbf{embs} = \lambda_1 \mathbf{embs}_{\text{src}} + \lambda_2 \mathbf{embs}^* \tag{5}$$

where λ_1 and λ_2 are tunable hyperparameters that control the balance between identity preservation and age transformation. By adjusting λ_1 and λ_2, we can achieve different levels of modification, allowing for flexible trade-offs between fidelity to the original identity and the desired age-related changes.

Finally, we use $\mathbf{embs}$ to generate personalized face image.

B.3. IP-Adapter-FaceID + DreamBooth

We integrate the two aforementioned approaches. First, following the training protocol outlined in Appendix A.1, we employ DreamBooth to train a personalized aging LoRA model. Subsequently, during inference, we utilize the IP-Adapter-FaceID method, which synthesizes target-age facial images by leveraging the prompt "`a photo of xx years old sks man/woman`" in conjunction with face embeddings $\mathbf{embs_{src}}$ obtained from a face encoder.

C. Evaluation Metrics Specification

In this work, we adopt several evaluation metrics to comprehensively assess the performance of our face generation system. These metrics not only quantify the quality of the generated images but also measure the degree of successful attribute transfer. Below, we elaborate on the motivation, formulation, and significance of each metric.

C.1. Face Similarity (Face Sim.)

Identity preservation is a critical aspect of face generation. To evaluate how well the generated face maintains the identity of the source, we extract 512-dimensional feature vectors from a pretrained FaceNet model [6]. Let $\mathbf{Emb}_{\text{src}}$ denote the source face embedding and $\mathbf{Emb}_{\text{tgt}}$ the generated face embedding. Their cosine similarity is computed as:

$$\text{Face Sim.} = \frac{\mathbf{Emb}_{\text{src}} \cdot \mathbf{Emb}_{\text{tgt}}}{\|\mathbf{Emb}_{\text{src}}\| \, \|\mathbf{Emb}_{\text{tgt}}\|} \tag{6}$$

This value ranges from -1 to 1, with higher values indicating better identity preservation. This metric is motivated by the need to ensure that the generated face reliably reflects the source identity, which is paramount for applications such as personalized avatar generation and forensic analysis.

C.2. Kernel Inception Distance (KID)

KID is employed to evaluate the overall quality and diversity of the generated images by comparing the feature distributions of real and generated samples. It is based on the Maximum Mean Discrepancy (MMD) computed with a radial basis function (RBF) kernel:

$$k(x, y) = \exp\left(-\gamma \|x - y\|^2\right)$$

with the default bandwidth parameter $\gamma = \frac{1}{512}$. Formally, KID is defined as:

$$\begin{aligned} \text{KID} = & \, \mathbb{E}_{\mathbf{Emb}_{\text{real}}, \mathbf{Emb}'_{\text{real}}} \left[k(\mathbf{Emb}_{\text{real}}, \mathbf{Emb}'_{\text{real}}) \right] \\ & + \mathbb{E}_{\mathbf{Emb}_{\text{gen}}, \mathbf{Emb}'_{\text{gen}}} \left[k(\mathbf{Emb}_{\text{gen}}, \mathbf{Emb}'_{\text{gen}}) \right] \\ & - 2\, \mathbb{E}_{\mathbf{Emb}_{\text{real}}, \mathbf{Emb}_{\text{gen}}} \left[k(\mathbf{Emb}_{\text{real}}, \mathbf{Emb}_{\text{gen}}) \right] \end{aligned} \tag{7}$$

Lower KID values indicate a smaller discrepancy between real and generated image distributions, reflecting higher generation quality. The motivation behind using KID lies in its unbiased estimation of distributional similarity in high-dimensional feature spaces.

C.3. Age Mean Absolute Error (AgeMAE)

Age transformation is a key attribute in our generation task. We quantify the accuracy of the age attribute by computing the Age Mean Absolute Error (AgeMAE). Let y_{target} denote the target age and $\hat{y}_{\text{gen}}$ be the predicted age of the generated face as obtained from [5]. The AgeMAE is calculated as:

$$\text{AgeMAE} = \frac{1}{N} \sum_{i=1}^{N} \left| \hat{y}_{\text{gen}}^{(i)} - y_{\text{target}}^{(i)} \right| \tag{8}$$

This metric directly reflects the model's capability to achieve the desired age transformation, with lower values indicating a closer match to the target age.

C.4. Relative Age Mean Absolute Error (R-AgeMAE)

To further assess the effectiveness of the Age Prompt in transferring the target age to the generated face, **we propose the Relative Age Mean Absolute Error (R-AgeMAE).** This metric measures the absolute difference between the average age obtained from the Age Prompt and the estimated age of the generated face. Let $\bar{y}_{\text{prompt}}$ denote the average predicted age from the Age Prompt (in cases where multiple images are used) and $\hat{y}_{\text{tgt}}$ the predicted age of the generated face. Then, R-AgeMAE is defined as:

$$\text{R-AgeMAE} = |\bar{y}_{\text{prompt}} - \hat{y}_{\text{tgt}}| \tag{9}$$

A lower R-AgeMAE indicates that the Age Prompt has been successfully transferred, as the generated face's age closely aligns with the intended prompt age. This metric is crucial for verifying the effectiveness of age conditioning in our generation framework.

C.5. Face Quality Assessment

In addition to the above metrics, we assess the visual quality of the generated images using a face quality assessment method [4]. This algorithm evaluates attributes such as sharpness, illumination, and overall visual fidelity, providing a quality score that objectively reflects the perceptual quality of the image. Higher quality scores indicate superior visual quality. The motivation for this metric is to ensure that the generative model not only preserves identity and age attributes but also produces aesthetically pleasing and realistic images.

C.6. Inference Time

Given that different generative methods may have varying computational demands, we also measure the inference time required to generate an image. This metric is of particular importance in practical applications where efficiency and scalability are critical. Shorter inference times imply that the model is more suitable for real-time or large-scale deployment, without compromising generation quality.

C.7. User Survey

Finally, to complement the objective metrics, we conduct user surveys to collect subjective evaluations of the generated images. The user survey results provide valuable insights into the perceived quality of the generated faces and help validate the effectiveness of our method from a human perspective. The user study consists of the following evaluation tasks:

(1) **Facial consistency**: Participants are presented with a source image and multiple target images generated by different methods. They are asked to select the target image that most closely resembles the source image.

(2) **Age prompt similarity**: Given a source image, an age prompt, and a set of target images, participants choose the target image that best matches the given age prompt.

(3) **Visual quality**: Participants receive both source and target images and are asked to select the target image that appears more photorealistic and exhibits a more natural age transformation.

D. Training Details

D.1. Training Dataset

During the construction of the second-stage training dataset, we applied a rigorous filtering process. We employed the YOLOv8 [3] object detection model to exclude images containing two or more faces and utilized the CLIB-FIQA face quality assessment method [4] to filter out low-quality images affected by blurriness, extreme lighting conditions, or corruption. To enable dynamic sampling, we applied an age estimation network [5] to predict the age of all images. Subsequently, we curated a cross-age identity pool from VGGFace2-HQ [1], ensuring that for each identity, a balanced number of images were retained across different age groups. This guarantees that during sampling, images from different age groups have approximately equal probabilities of being selected, thus maintaining diversity and fairness in training. Finally, we leveraged Human-LLaVA [2] to generate detailed textual descriptions for each face image.

D.2. Training Details

We use Stable Diffusion v1.5 as the base model for training. To achieve better inference results and demonstrate the plug-and-play feature of the age adapter, we set Realistic_Vision_V4.0 as our base model for inference. Our baseline method also uses Realistic_Vision_V4.0 as the base model.

In the Disentangled Representation Learning Stage, we freeze the ViT backbone and train only the neck and head networks. In the Face Reconstruction Stage, the encoder trained in the Disentangled Representation Learning Stage is frozen, while the focus shifts to training the Age Adapter. Additionally, LoRA modules are utilized to fine-tune the attention modules of the U-Net.

In the Disentangled Representation Learning Stage of the loss function are set as follows:

$$\lambda_{\text{age}} = 10^{-2}, \quad \lambda_{\text{advid}} = 10^{-3}, \quad \lambda_{\text{advage}} = 2 \times 10^{-3}$$

The batch size is set to 1024, and training is conducted for approximately 30K steps.

In the Face Reconstruction Stage of the training strategy are set as follows:

$$p_s = 0.05, \quad p_a = 0.05, \quad p_d = 0.05, \quad lr = 10^{-4}.$$

In the Face Reconstruction Stage, all training images are downsampled to 256×256, with a batch size of 24 per GPU. The training is performed on five NVIDIA 3090 GPUs for approximately 200K steps.

E. More Experiments

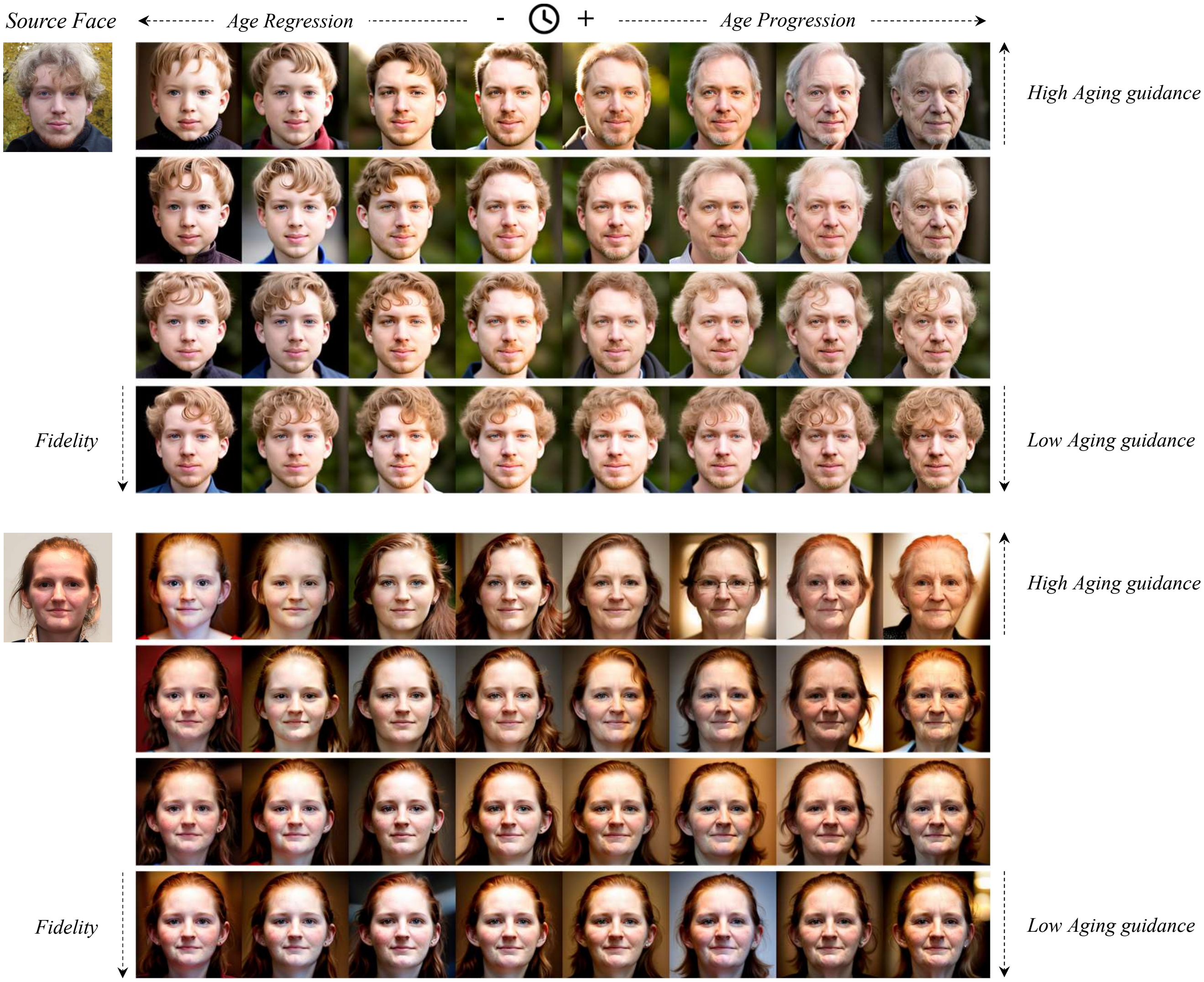

Figure 1: **Perform lifespan synthesis using different aging guidance.** As aging guidance increases, the generated faces exhibit more pronounced aging traits, such as deeper wrinkles and looser skin. When aging guidance decreases, the age features gradually fade, creating a natural transition between the source image and age prompts. Spherical interpolation enables fine-grained control over aging features, ensuring smooth and diverse age transformations.

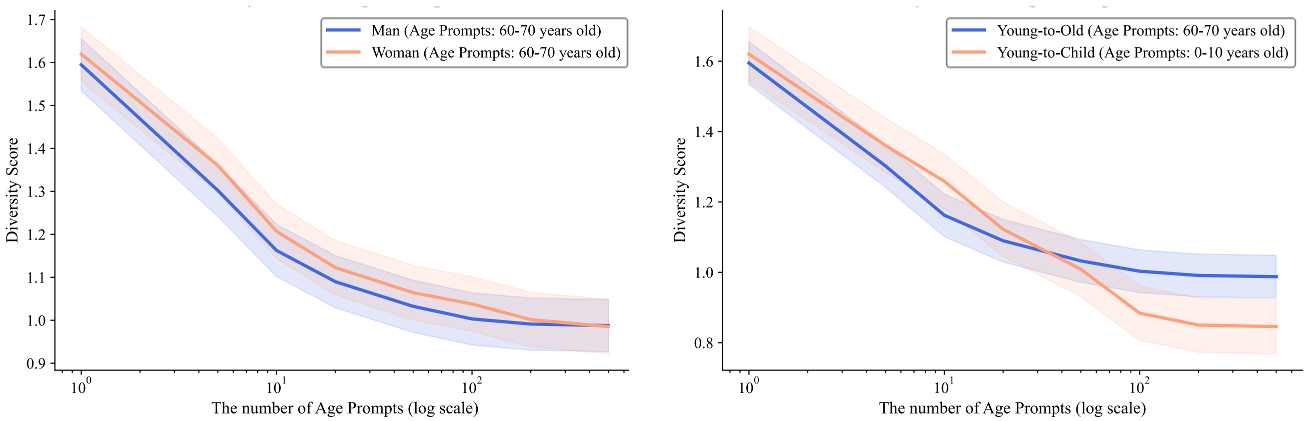

Figure 2: **The impact of age prompt quantity on generated results,** showing a trend consistent with Fig. **??**.

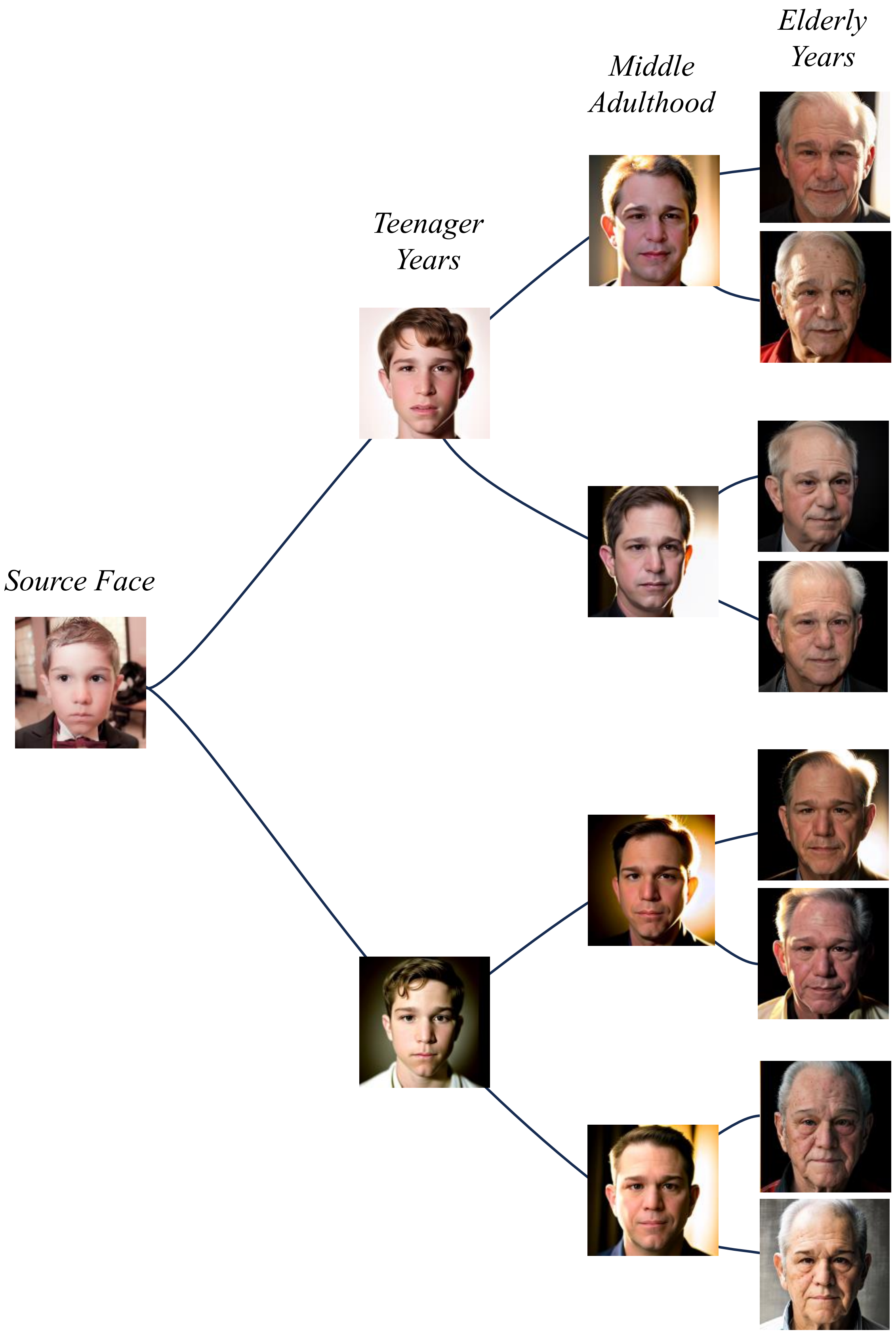

Figure 3: **Additional Face Aging Results.** Our method generates diverse age transformation results, forming a tree-like structure, as shown in the figure. This hierarchical representation demonstrates the model's ability to adapt to various age characteristics while preserving identity, producing diverse and coherent outputs.

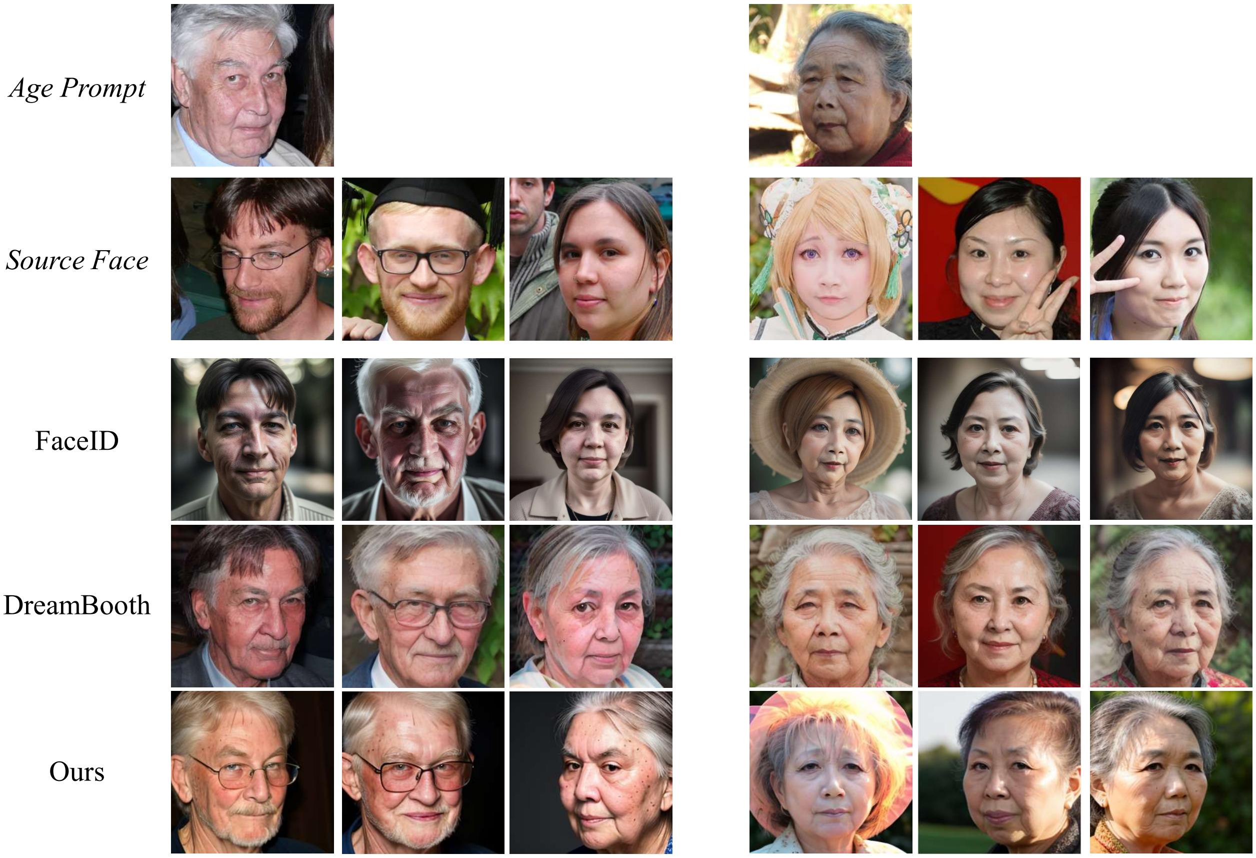

Figure 4: **Additional qualitative comparison with baseline methods.** Note the results generated by our method: In the left image, the generated face is consistent with the Age Prompt, exhibiting age spots. In the right image, the generated face shows signs of drooping eyelids. The fidelity is significantly higher than that of the baseline methods.

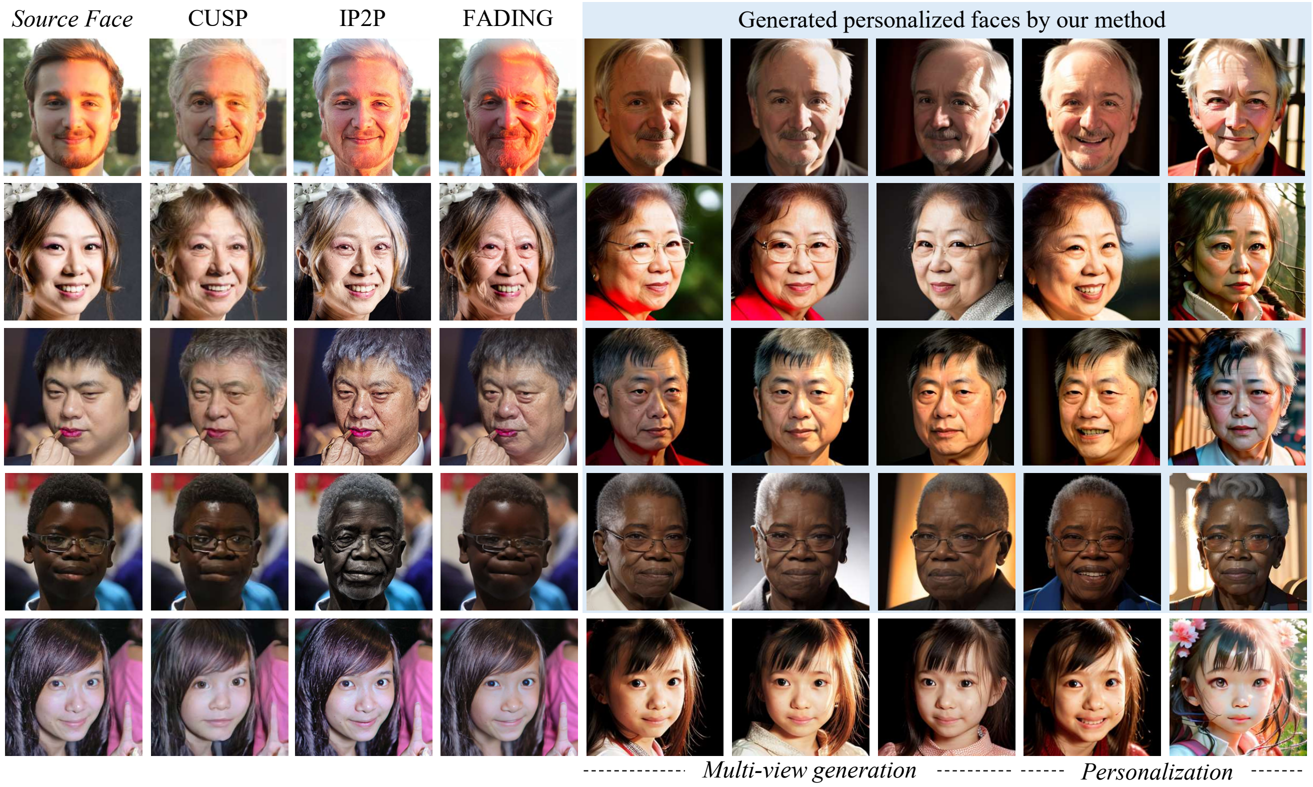

Figure 5: **Additional qualitative comparison of face aging.** Our method offers greater flexibility, supporting multi-view face generation and enabling precise facial expression manipulation through text prompts. Additionally, it can be combined with stylized LoRA to generate anime-style faces.

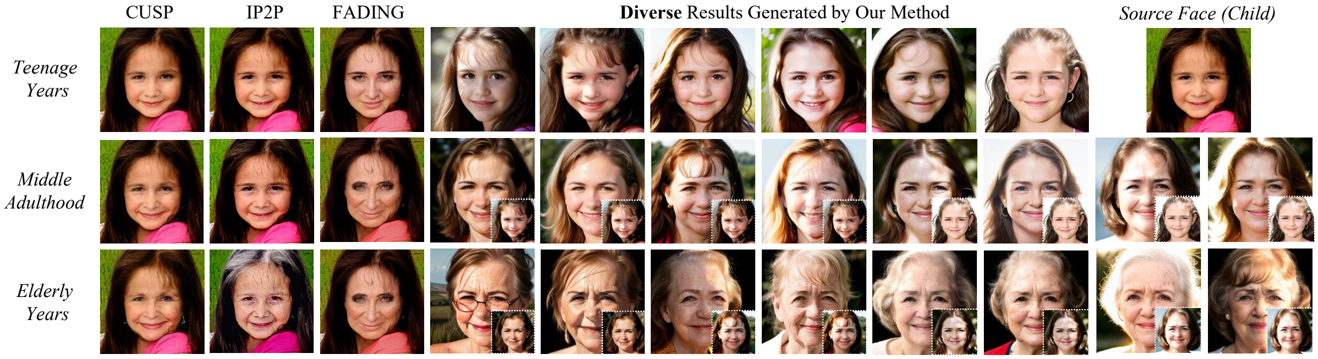

Figure 6: **Additional qualitative comparison of face aging.** Compared to comparison methods, TimeBooth generates faces that are both more natural and diverse, effectively avoiding the issue of ’child faces with elderly textures.’

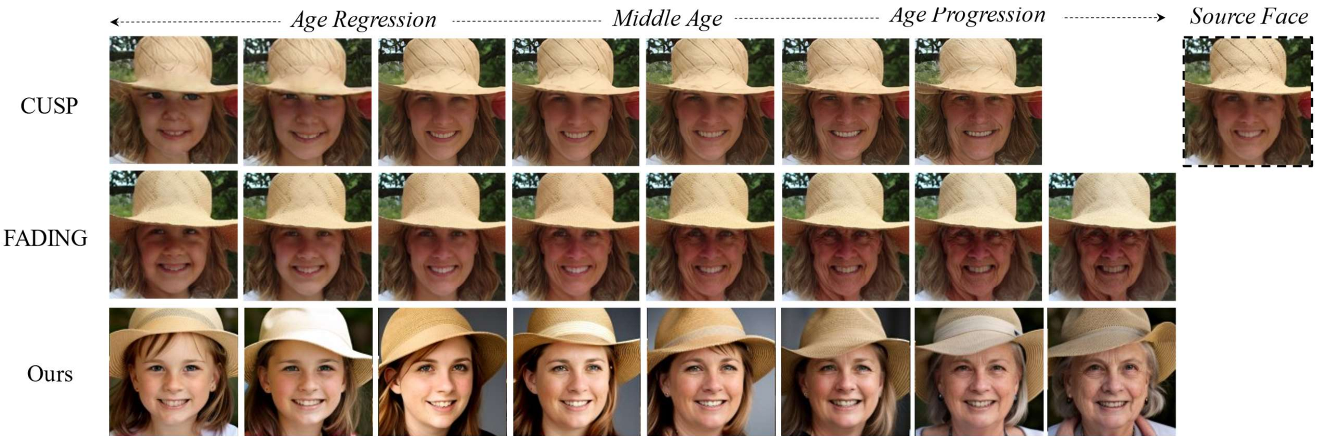

Figure 7: **Qualitative comparison of lifespan synthesis.** TimeBooth generates more natural images with high fidelity and consistency.

F. Model Details

F.1. Disentangled Face Encoder

Layer	Operation	Params	Input	Output
proj_in	Linear	512 → 512	(B, S, 512)	(B, S, 512)
Layer 1-4 (Attn)	PerceiverAttention	8 heads, 64 dim/head	Query: (B,16,512) Key/Val: (B,S,512)	(B,16,512)
Layer 1-4 (FFN)	FeedForward	512→2048→512	(B,16,512)	(B,16,512)
proj_out	Linear	512 → 512	(B,16,512)	(B,16,512)
norm_out	LayerNorm	-	(B,16,512)	(B,16,512)

Table 1: The network architecture of the neck.

Layer	Type	Parameters	Input Shape	Output Shape
norm_mlp	LayerNorm	-	(B,16,512)	(B,16,512)
flatten	Flatten	-	(B,16,512)	(B,8192)
linear1	Linear	8192→2048	(B,8192)	(B,2048)
act1	LeakyReLU	-	(B,2048)	(B,2048)
linear2	Linear	2048→512	(B,2048)	(B,512)
act2	LeakyReLU	-	(B,512)	(B,512)
dropout	Dropout	p=0.5	(B,512)	(B,512)
age_prob_head	Linear	512→101	(B,512)	(B,101)

Table 2: The network architecture of the age head.

Layer	Type	Parameters	Input	Output
LayerNorm	Normalization	-	(B,16,512)	(B,16,512)
Flatten	Reshape	-	(B,16,512)	(B,8192)
Linear1	Projection	8192→4096	(B,8192)	(B,4096)
LeakyReLU1	Activation	α=0.01	(B,4096)	(B,4096)
Linear2	Projection	4096→2048	(B,4096)	(B,2048)
LeakyReLU2	Activation	α=0.01	(B,2048)	(B,2048)
Dropout	Regularization	p=0.5	(B,2048)	(B,2048)
Linear3	Projection	2048→512	(B,2048)	(B,512)

Table 3: The network architecture the ID head. The configuration of CosFace is omitted here. Please refer to [7].

F.2. Age Adapter

Layer Type	**Parameters**	**Input Shape**	**Output Shape**
LayerNorm	dim=768	(B, N, 768)	(B, N, 768)
LayerNorm	dim=768	(B, N, 768)	(B, N, 768)
LayerNorm	dim=768	(B, N, 768)	(B, N, 768)
Linear + GELU + Linear	512 → 1024 → 768	(B, N, 512)	(B, N, 768)
Linear + GELU + Linear	512 → 1024 → 768	(B, N, 512)	(B, N, 768)
Cross Attention Blocks(ID→Age)	depth=4, heads=12	(B, N, 768)	(B, N, 768)
Cross Attention Blocks(Age→ID)	depth=4, heads=12	(B, N, 768)	(B, N, 768)
Concatenation	-	(B, N, 768) × 2	(B, 2N, 768)
LayerNorm	dim=768	(B, 2N, 768)	(B, 2N, 768)
Learnable Tokens	32 tokens	(32, 768)	(B, 32, 768)
Q-Former	depth=8, heads=12	(B, 32, 768), (B, 2N, 768)	(B, 32, 768)

Table 4: The network architecture of Age-Conditioned ID Encoder

References

[1] Xuanhong Chen, Bingbing Ni, Yutian Liu, Naiyuan Liu, Zhilin Zeng, and Hang Wang. Simswap++: Towards faster and high-quality identity swapping. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):576–592, 2023.

[2] Dawei Dai, Xu Long, Li Yutang, Zhang Yuanhui, and Shuyin Xia. Humanvlm: Foundation for human-scene vision-language model, 2024.

[3] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLO, Jan. 2023.

[4] Fu-Zhao Ou, Chongyi Li, Shiqi Wang, and Sam Kwong. Clib-fiqa: face image quality assessment with confidence calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1694–1704, 2024.

[5] Lixiong Qin, Mei Wang, Chao Deng, Ke Wang, Xi Chen, Jiani Hu, and Weihong Deng. Swinface: A multi-task transformer for face recognition, expression recognition, age estimation and attribute estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(4):2223–2234, 2023.

[6] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[7] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274, 2018.