

StreamAvatar: Streaming Diffusion Models for Real-Time Interactive Human Avatars

Supplementary Material

6. Additional Details of the Teacher Model

Base Model Architecture. Our teacher model is built upon Wan2.2-TI2V-5B [47], a Rectified Flow [25] model comprising a causal video VAE and a bidirectional DiT denoiser. The VAE compresses video data into a compact latent space with a spatial downsampling factor of $16\times$ along both height and width, and a temporal downsampling factor of $4\times$, thereby substantially reducing the computational cost of the DiT. Concretely, a video of n frames is encoded into $\lfloor (n-1)/4 \rfloor + 1$ latent frames, where the first frame is encoded independently (i.e., without temporal compression) and subsequent frames are compressed at a ratio of 4. All generation and denoising operations are performed in this latent space.

The DiT takes as input a text prompt, a noisy video latent, an optional clean reference first frame, and a diffusion timestep, and predicts the Rectified Flow velocity field for denoising. Reference-image guidance is realized by replacing the first frame of the noisy video latent with the clean, noise-free latent of the reference image; the bidirectional self-attention mechanism then propagates identity and appearance information from this anchor frame to all subsequent frames and achieve spatial and temporal coherence. Cross-attention layers inject the text-prompt information into the video latent to enable text-based control.

Audio Encoder. To obtain audio features suitable for injection into the video model as driving conditions, we design an audio encoder as illustrated in the yellow region of Fig. 4. We extract multi-layer deep features from a pretrained Wav2Vec 2.0 [1] encoder. Because the VAE’s temporal compression ratio differs between the first frame and subsequent frames, an explicit step is required to align Wav2Vec features with VAE latent frames.

We adopt a context-window approach: each latent frame attends to a short temporal neighborhood of Wav2Vec features centered around its corresponding video frame, so that anticipatory and carry-over acoustic cues (e.g., mouth opening before speech onset, or a prolonged sigh) are captured. Denoting the Wav2Vec feature corresponding to the uncompressed video frame i as f_i (with $f_i = f_0$ for $i < 0$), the audio feature f'_t assigned to latent frame t is defined as:

$$f'_t = \begin{cases} \text{concat}(\{f_i\}_{i=t-2}^{t+2}), & t = 0, \\ \text{concat}(\{f_i\}_{i=4t-5}^{4t+2}), & t > 0. \end{cases} \quad (1)$$

Because f'_0 aggregates 5 Wav2Vec frames while f'_t ($t > 0$) aggregates 8, their raw dimensions differ. We therefore apply

separate lightweight MLP projectors (denoted “Audio Proj.”) to map both cases to a common feature dimension. Importantly, the talking and listening streams use *independent* projectors, allowing each to learn phase-specific representations. After projection, the features are concatenated along the temporal axis to form frame-aligned audio feature sequences, yielding the talking audio feature sequence $\{a_{\text{talk},t}\}$ or the listening audio feature sequence $\{a_{\text{listen},t}\}$.

Audio Attention Modules. As shown in the cyan region of Fig. 4, to enable speech-driven generation of both talking and listening motions, we insert audio attention modules into each of the $N_{\text{blk}} = 30$ DiT blocks of the video model, injecting audio information via cross-attention between the video latents and the audio features. To preserve strict temporal correspondence between audio and motion, each latent frame’s query tokens attend *only* to the audio features assigned to that same frame, rather than to the full audio sequence. This frame-wise cross-attention design prevents temporal leakage and ensures that lip movements and gestures remain tightly synchronized with the driving audio. All other layers in the Transformer block (self-attention, text cross-attention, and feed-forward layers) remain unmodified and audio-agnostic.

Training Procedure. We train the teacher model following the standard Rectified Flow training paradigm under the Flow Matching [23] framework. Given a video latent x^0 and its corresponding conditions (reference first frame, audio features, and text prompt), we sample a random timestep $n \in (0, 1)$ and construct a noisy latent x^n by linearly interpolating between x^0 and Gaussian noise ϵ : $x^n = (1-n)x^0 + n\epsilon$. The model is trained with a mean squared error loss to predict the velocity field $v = \epsilon - x^0$ at the sampled point. During training, we adopt a two-stage strategy: we first freeze the pretrained DiT and only train the audio projection and audio attention modules, then unfreeze the DiT and fine-tune all parameters jointly.

7. Additional Experiments

7.1. Quantitative Comparison/Ablation on EMTD

To further evaluate our approach, we compare it with baseline methods and ablation variants on the EchoMimicV2 Testing Dataset (EMTD) [33]. The EMTD dataset contains 110 front-facing, half-body speech videos. Quantitative results are presented in Tab. 4. Our method outperforms all comparison methods across almost all metrics, and the ablation results further demonstrate the effectiveness of our design.

Table 4. Quantitative comparison with SoTA talking avatar video generation methods on the EMTD dataset. Best in **bold** and second best underlined.

Method	FID	FVD	IQA	ASE	Sync-C	Sync-D	HKV	HA
StableAvatar	91.63	840.86	3.67	2.37	3.04	12.16	57.99	0.794
OmniAvatar	75.20	982.09	3.72	2.45	<u>7.68</u>	<u>7.97</u>	29.04	0.889
HY-Avatar	<u>63.09</u>	<u>765.05</u>	3.91	2.57	7.35	8.36	66.07	0.880
Hallo3	91.15	898.19	3.57	2.26	5.62	9.72	29.52	0.874
EchoMimicV3	67.35	822.89	<u>3.98</u>	<u>2.68</u>	3.00	12.20	56.58	<u>0.921</u>
Ours (baseline)	107.50	1254.66	3.25	2.12	7.23	8.26	80.15	0.934
+ref sink	81.10	1060.86	3.69	2.36	7.60	8.05	79.67	0.908
+RAPR	63.71	801.68	4.03	2.66	7.45	8.04	62.09	0.925
+GAN w/o D_{CA}	59.87	749.32	4.03	2.64	7.67	7.88	36.79	0.929
Ours	61.84	683.14	4.13	2.78	8.06	7.93	<u>62.60</u>	0.935

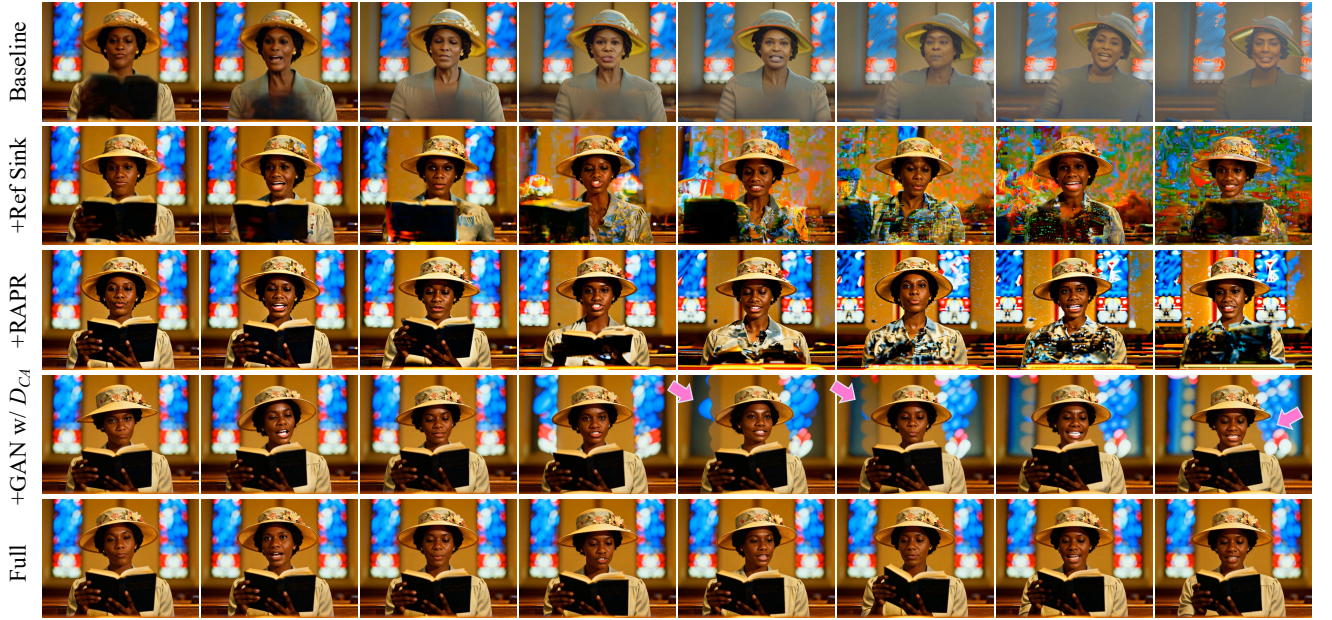


Figure 7. Qualitative Ablation Results.

7.2. Qualitative Ablation Results

We present qualitative ablation results in Fig. 7 and the demo video. Note how the addition of key components gradually improves long-term generation stability and consistency, identity preserving, and visual quality.

7.3. User Study

We conduct a user study to comprehensively evaluate our method. Participants are shown paired video clips generated by our approach and a comparison method, and asked to assess the two along five dimensions: audio–lip synchronization (Sync), motion dynamics (Dynamics), temporal continuity and smoothness (Continuity), visual quality and naturalness (Quality), and identity preservation (Identity).

For each pair, participants indicate whether they prefer our method, prefer the comparison method, or have no preference. In total, we collect 960 paired comparisons from 24 participants. As illustrated in Tab. 5, our method consistently outperforms the state-of-the-art baselines across almost all comparisons, which aligns closely with our quantitative evaluation.

7.4. Comparison with Interactive Head Generation

We compare our method with state-of-the-art interactive head generation methods, including INFP [67] and ARIG [11]. As these methods are not open-sourced, we conduct qualitative comparison with the results from their project pages, as shown in Fig. 8 and the demo video. Although our model is designed and trained for body video generation, it still

Table 5. **User study results.** The table presents the pairwise preference rates (%) across different metrics, formatted as (Ours / Baseline). Winning values are highlighted in **bold**. The remaining percentage in each comparison accounts for “Tie” (no preference) cases.

Ours vs X	Sync (%)	Quality (%)	Dynamics (%)	Identity (%)	Continuity (%)
EchoMimicV3	91.4 / 2.1	68.6 / 4.9	74.1 / 8.6	47.6 / 1.0	50.8 / 1.1
Hallo3	86.2 / 2.7	79.9 / 2.6	47.1 / 28.6	64.6 / 1.0	68.3 / 1.5
HY-Avatar	41.2 / 18.0	48.5 / 13.4	16.5 / 57.2	28.9 / 7.7	44.3 / 8.8
OmniAvatar	45.9 / 14.8	53.6 / 6.1	75.0 / 13.3	25.5 / 4.1	28.1 / 6.1
StableAvatar	74.0 / 5.6	65.3 / 5.1	39.3 / 36.7	61.7 / 3.6	66.8 / 4.6

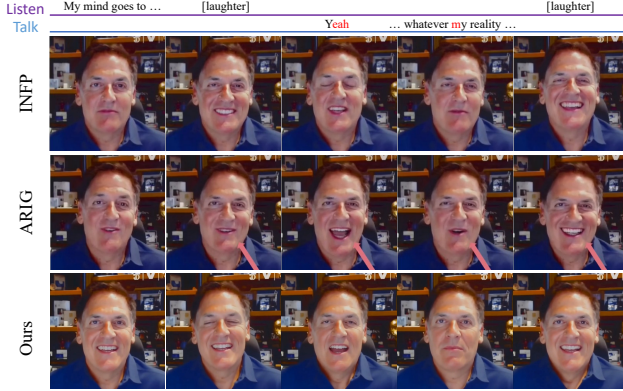


Figure 8. Qualitative comparison with SoTA interactive head generation methods. Please ignore the arrows which come with the original video on ARIG’s project page.

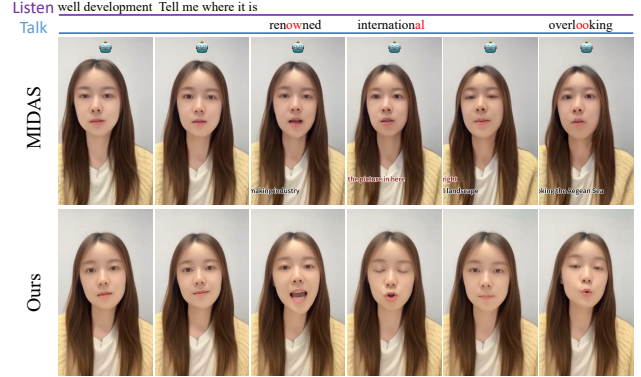


Figure 9. Qualitative comparison with MIDAS [3].

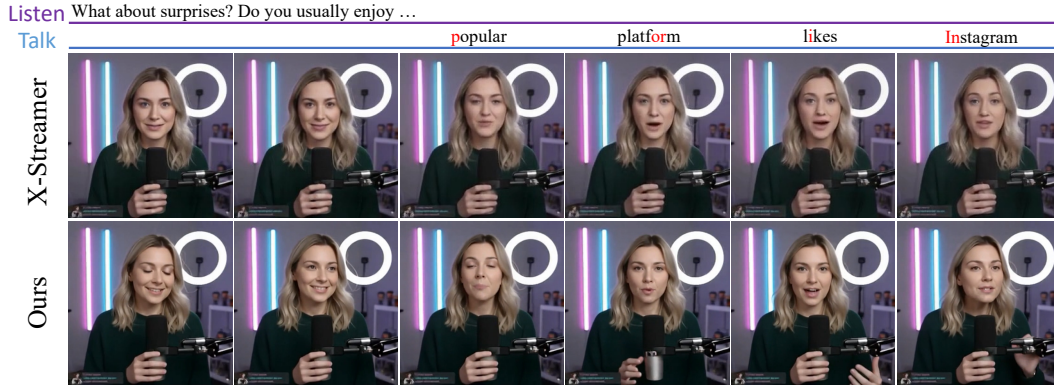


Figure 10. Qualitative comparison with X-Streamer [51].

performs on par with these dedicated head avatar methods, while delivering the best visual quality.

7.5. Comparison with Streaming Interactive Avatar Generation

We further compare our method with current state-of-the-art streaming interactive avatar generation methods, including MIDAS [3] and X-Streamer [51]. As these methods are not open-sourced, we conduct qualitative comparison with the results from their project pages, as shown in Fig. 9, Fig. 10, and the demo video. Our method produces more accurate lip

synchronization and more vivid expressions than MIDAS. It is also worth noting that our method is *one-shot*, whereas MIDAS requires person-specific finetuning. Our method also exhibits more natural listening behaviors, more diverse motions, and higher visual quality than X-Streamer.

7.6. Long Video Generation

Thanks to the streaming architecture with all our proposed techniques to improve consistency and stability for long video generation, our approach can generate videos of arbitrary length without quality degradation in real-time. Fig. 11

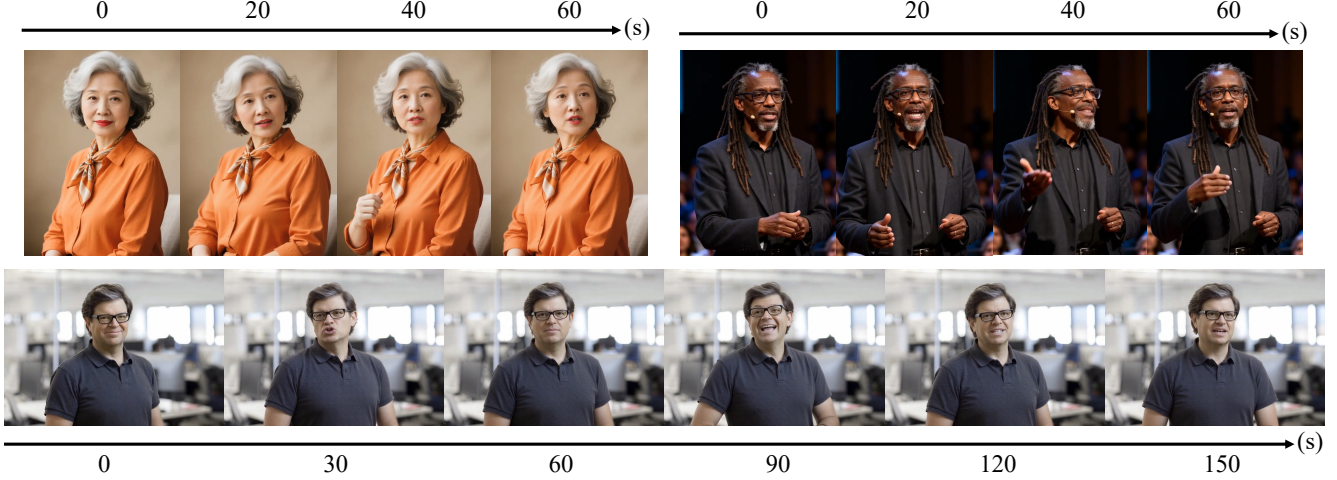


Figure 11. Long video generation results.

shows sampled frames from generated videos of up to 150 seconds. The results maintain consistent identity, appearance, and visual quality throughout the entire duration. We refer readers to the supplementary demo video for the full-length results.

8. Runtime Details

Table 6. Evaluation of the real-time performance of our model.

Module	RTF	FFD
DiT	0.69	0.33s
VAE	0.82	0.39s

Our model generates videos at 25 FPS. To enable real-time generation, we distribute the DiT denoising and VAE decoding processes across two NVIDIA H800 GPUs. We evaluate the performance under our default model configuration (denoising steps $N=3$, chunk size $C=3$, total KV cache length $L=10$) when generating videos at a resolution of 928×704 . We report two metrics: the Real-Time Factor (RTF), defined as the ratio between the inference time and the duration of the generated video segment, and the First Frame Delay (FFD), defined as the time elapsed from receiving the input to producing the first output frame.

The results are listed in Tab. 6. Since the RTF values of all modules are below 1, the system supports real-time generation.

Audio Lookahead. Recall from the audio feature extraction in Eq. (1) that the context window for latent frame $t > 0$ includes two “future” Wav2Vec features f_{4t+1} and f_{4t+2} , which correspond to video frames that have not yet been observed at generation time. We empirically find that replacing these two features with f_{4t} (i.e., repeating the current-frame

feature) has a negligible effect on generation quality. Consequently, in the deployed system the model does not need to wait for any future audio input beyond the current chunk boundary. The overall system latency is therefore given by the sum of the FFD and the input chunk buffering delay ($C \times 4/25 = 0.48\text{s}$), yielding a total end-to-end latency of approximately 1.20s.

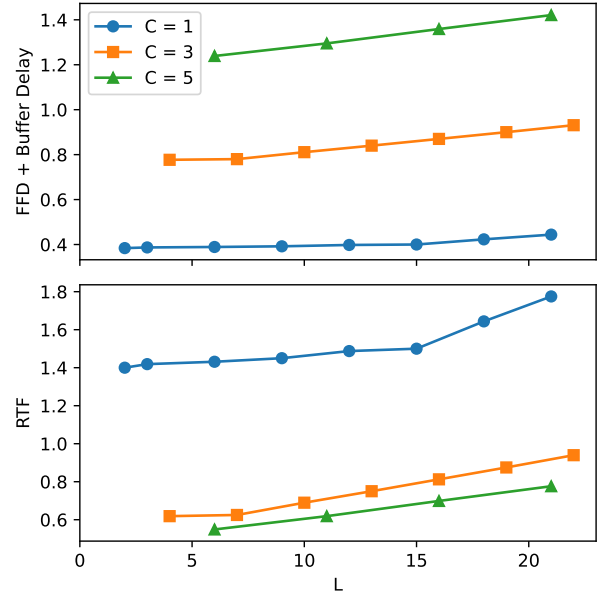


Figure 12. Impact of chunk size C and KV cache total length L on real-time performance. **Top:** First Frame Delay (FFD). **Bottom:** Real-Time Factor (RTF).

Impact of Chunk Size and KV Cache Length. We further analyze how the chunk size C and the total KV cache length L affect real-time performance, as illustrated in Fig. 12. A

smaller C reduces the first-frame delay because each chunk covers fewer frames; however, it also increases the per-frame overhead and decreases the intra-chunk parallelism, pushing RTF above 1 and breaking the real-time constraint. Conversely, a larger C improves throughput (lower RTF) at the cost of higher latency. Increasing L provides the model with a longer temporal context, which benefits temporal consistency, but also raises both latency and RTF because each attention operation must attend to more cached tokens. Too small an L , on the other hand, degrades temporal coherence as the model loses access to sufficient history. Our default configuration ($C=3$, $L=10$) strikes a balanced trade-off among latency, throughput, and generation quality.

9. Ethical Considerations

This work focuses on talking avatar generation for constructive, human-centered applications, and is not intended to support deceptive or harmful media. As with any generative technology, misuse is possible, such as creating fraudulent identities, fabricating false narratives, or generating avatars for harassment. To mitigate these risks, we commit to safeguards including embedding watermarks and clearly disclosing that all outputs are synthetic when deploying the technology. We also aim to collaborate with the research community to develop improved deepfake detection tools and support efforts to establish standards for media provenance.