

# Avatar Forcing: Real-Time Interactive Head Avatar Generation for Natural Conversation

## Supplementary Material

**Organization** The appendix is organized as follows: In Sec. A, we provide the additional backgrounds of our work. We describe the details of our model architecture in Sec. B and the preference optimization method in Sec. C. Experimental details are presented in Sec. D, and further discussion is provided in Sec. E.

### A. Background

**Flow Matching** Flow matching [12, 13] is a generative model that transforms a simple prior distribution  $p_0$ , for example, a Gaussian distribution, into the target data distribution  $p_1$  via an ordinary differential equation (ODE):

$$\frac{dx_t}{dt} = v_\theta(x_t, t), \quad t \in [0, 1], \quad (8)$$

where for fixed  $x_0 \sim p_0$  and  $x_1 \sim p_1$ , the intermediate sample is a linear interpolation  $x_t = tx_1 + (1 - t)x_0$ . The training objective of flow matching is to regress the vector field  $v_\theta$  toward the target vector field  $v_t = x_1 - x_0$ :

$$\mathcal{L}_{FM}(\theta) = \mathbb{E}_{t, x_t} [\|v_\theta(x_t, t) - (x_1 - x_0)\|]. \quad (9)$$

It then generates target samples by solving Eq. (8). Note that flow matching can be interpreted as a diffusion model [9, 16] where a noise schedule follows the linear trajectory between the prior and the target data.

### B. Details on Model Architecture

In Sec. B.1, we provide more details on the motion latent auto-encoder. In Sec. B.2, we provide more details on the vector field predictor  $v_\theta$ .

#### B.1. Motion Latent Auto-encoder

In Fig. 9, we show an overview of the motion latent auto-encoder. It encodes an image into a latent vector that can be decomposed into an identity representation (i.e., appearance) and the motion representation. This auto-encoder is trained to reconstruct a driving image using a source image that shares the same identity. During each training iteration, the encoder encodes two images  $S$  and  $D$  drawn from the same video clip, and computes the  $z_{S \rightarrow D} := z_S + \mathbf{m}_D$  that transforms the source image into the reconstructed  $\hat{D}$ . This explicit decomposition yields a compact motion representation, enabling fast motion generation.

We train this auto-encoder on our dataset, following the training objective described in the original paper. For more details, including the property of this latent space, training details, please refer to [11, 20].

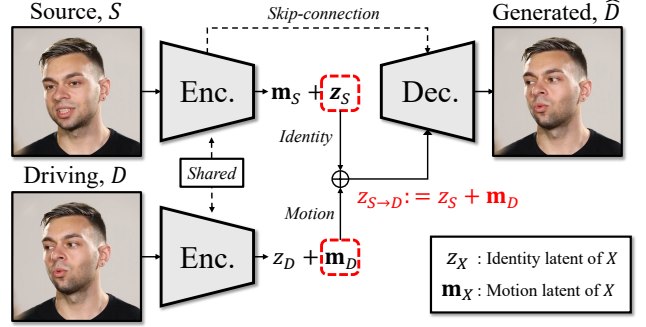


Figure 9. **Overview of Motion Latent Encoder.** It encodes an image into a latent vector that has explicit identity-motion decomposition.

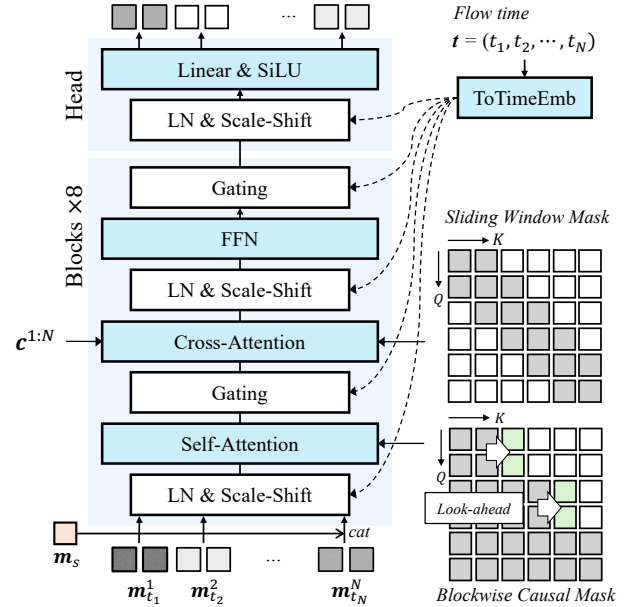


Figure 10. **Detailed architecture for Motion Generator in  $v_\theta$ .**

#### B.2. Vector Field Model $v_\theta$

**Model Architecture** The model  $v_\theta$  comprises two main components: the Dual Motion Encoder and the Causal DFoT Motion Generator. The Dual Motion Encoder unifies three multimodal inputs through a cross-attention layer, computed as  $\text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V$ , where

$$Q = \mathbf{q}W_q, \quad K = \mathbf{k}W_k, \quad V = \mathbf{v}W_v, \quad (10)$$

and  $W_q, W_k, W_v \in \mathbb{R}^{d \times d}$  are learnable projection ma-

trices for the query  $\mathbf{q}$ , key  $\mathbf{k}$ , and value  $\mathbf{v} \in \mathbb{R}^{N \times d}$ , respectively ( $N$  is the number of latents). In the first cross-attention layer, the encoder captures holistic verbal and non-verbal user motion by using the user motion latent  $\mathbf{m}_u$  as the query. In the second layer, it integrates this aligned user motion with the avatar’s audio by taking the avatar audio as the query. We use four attention heads, each with a hidden dimension of  $d = 512$ , for both cross-attention layers.

In Fig. 10, we provide a detailed architecture for the motion generator. It consists of eight DFoT transformer blocks followed by a transformer head. Specifically, in each DFoT block, noisy latents  $(\mathbf{m}_{t_1}^1, \mathbf{m}_{t_2}^2, \dots, \mathbf{m}_{t_N}^N)$  are modulated by the flow time  $\mathbf{t} = (t_1, t_2, \dots, t_N)$  through a shared AdaLN scale-shift coefficients (ToTimeEmb layer) [3].

For the attention modules, we use Blockwise Causal Look-ahead Mask (Eq. (5)) in self-attention and a Sliding-window Attention Mask for aligning the driving signal  $\mathbf{c}^{1:N}$  to the noisy latents. Specifically, we introduce the Blockwise Casual Look-ahead Mask to ensure the causal motion generation in our motion latent space, which significantly improves the temporal consistency of the generated video, as demonstrated in Sec. D.4. Unlike the recent video diffusion models that employ a spatio-temporal compression module [10, 21] where each latent correlates to multiple video frames by the compression rate (e.g.,  $4\times$  or  $8\times$ ), our motion latent has one-to-one correspondence with each frame in pixel space. Under this setting, a simple (block-wise) causal mask alone produces the temporal inconsistencies across the frames or blocks.

## C. Details on Preference Optimization

**Training Objective Formulation** Inspired by DiffusionDPO [19], we formulate the training objective  $\mathcal{L}_{DPO}$  in the context of diffusion forcing [2]. Let  $(\mathbf{m}^l, \mathbf{m}^w)$  denote a pair of less-preferred and preferred motion latents, each consisting of  $N$  frames, where  $\mathbf{m}^l := (\mathbf{m}^{l,n})_{n=1}^N$  and  $\mathbf{m}^w := (\mathbf{m}^{w,n})_{n=1}^N$ . Following the per-token independent noising process of diffusion forcing, we construct the noisy latent pairs as:

$$\begin{aligned} \mathbf{m}_{t_n}^{w,n} &:= t_n \mathbf{m}^{w,n} + (1 - t_n) \mathbf{m}_0^n, \\ \mathbf{m}_{t_n}^{l,n} &:= t_n \mathbf{m}^{l,n} + (1 - t_n) \mathbf{m}_0^n, \end{aligned} \quad (11)$$

where  $n \in [1, N]$  is the frame index,  $t_n \in [0, 1]$  is the  $n$ -th flow time, and  $\mathbf{m}_0 := (\mathbf{m}_0^n)_{n=1}^N \in \mathbb{R}^{N \times d}$  is the noise sequence. With these notations, we formulate  $\mathcal{L}_{DPO}$  as

$$\begin{aligned} \mathcal{L}_{DPO}(\theta) &= -\mathbb{E}_{n, t_n, \mathbf{c}^n, (\mathbf{m}^{w,n}, \mathbf{m}^{l,n})} \\ &\log \sigma \left( -\beta [\|v_{t_n}^{w,n} - v_\theta(\mathbf{m}_{t_n}^{w,n}, t_n, \mathbf{c}^n)\| \right. \\ &\quad - \|v_{t_n}^{w,n} - v_{\text{ref}}(\mathbf{m}_{t_n}^{w,n}, t_n, \mathbf{c}^n)\| \\ &\quad - (\|v_{t_n}^{l,n} - v_\theta(\mathbf{m}_{t_n}^{l,n}, t_n, \mathbf{c}^n)\| \\ &\quad \left. - \|v_{t_n}^{l,n} - v_{\text{ref}}(\mathbf{m}_{t_n}^{l,n}, t_n, \mathbf{c}^n)\|)] \right), \end{aligned} \quad (12)$$

## Algorithm 2 Motion inference with KV caching (Detailed)

---

**Require:** ODE timesteps  $\{t_n\}_{n=0}^T$ , motion generator  $v_\theta$ , video length  $N$ , block size  $B$ , lookahead size  $l$ , max cache size  $M$ , user inputs  $(\mathbf{a}_u, \mathbf{m}_u)$ , avatar audio  $\mathbf{a}^i$ , latent-to-frame decoder  $\text{Dec}$ , offset  $\mathbf{O}$ , and id latent  $z_S$ .

- 1: **Divide** the frames into  $L = \lceil N/B \rceil$  blocks
- 2: **Initialize**  $\mathbf{KV}, \mathbf{cKV} \leftarrow [], []$   $\triangleright$  Frame & condition caches
- 3: **for**  $i = 1$  to  $L$  **do**
- 4:   **Sample** Noise block  $\mathbf{m}_{t_0}^i \sim \mathcal{N}(0, \mathbf{I})$ ,  $\triangleright \mathbf{m}_{t_0}^i \in \mathbb{R}^{B \times d}$
- 5:   **Acquire** User inputs  $(\mathbf{a}_u^i, \mathbf{m}_u^i)$  and avatar audio  $\mathbf{a}^i$ .
- 6:   **Set**  $\mathbf{c}^i \leftarrow (\mathbf{a}_u^i, \mathbf{m}_u^i, \mathbf{a}^i)$   $\triangleright$  Condition triplet
- 7:   **Merge offset**  $\mathbf{m}_{t_0}^i, \mathbf{c}^i \leftarrow \text{Concat}(\mathbf{m}_{t_0}^i, \mathbf{c}^i; \mathbf{O}^i)$
- 8:   **for**  $j = 0$  to  $T$  **do**
- 9:     **Solve ODE:**  $\mathbf{m}_{t_{j+1}}^i \leftarrow v_\theta(\mathbf{m}_{t_j}^i, t_j; \mathbf{c}^i, \mathbf{KV}, \mathbf{cKV})$
- 10:   **end for**
- 11:   **Decode & Return**  $\mathbf{x}_1^i \leftarrow \text{Dec}(z_S, \mathbf{m}_1^i) \in \mathbb{R}^{B \times 3 \times H \times W}$
- 12:   **Update caches**  $\mathbf{kv}_i, \mathbf{ckv}_i \leftarrow v_\theta(z_1^i, 1; \mathbf{c}^i, \mathbf{KV}, \mathbf{cKV})$
- 13:   **if**  $|\mathbf{KV}| = |\mathbf{cKV}| = M$  **then**
- 14:      $\mathbf{KV}.\text{pop}(0)$  and  $\mathbf{cKV}.\text{pop}(0)$
- 15:   **end if**
- 16:    $\mathbf{KV}.\text{append}(\mathbf{kv}_i)$  and  $\mathbf{cKV}.\text{append}(\mathbf{ckv}_i)$
- 17:   **Update offset**  $\mathbf{O}^{i+1} \leftarrow (\mathbf{m}_1^i[-l:], \mathbf{c}^i[-l:])$
- 18: **end for**

---

where  $\mathbf{c}^n$  is the  $n$ -th unified condition,  $v_{\text{ref}}$  is the reference vector field model, the target vector fields for less-preferred and preferred samples are given by

$$v_{t_n}^{l,n} := \mathbf{m}_{t_n}^{l,n} - \mathbf{m}_0^n \quad \text{and} \quad v_{t_n}^{w,n} := \mathbf{m}_{t_n}^{w,n} - \mathbf{m}_0^n. \quad (13)$$

## D. Experimental Details

### D.1. Inference Details

**Classifier-Free Guidance (CFG)** We apply independent classifier-free guidance (CFG) [1] for multiple driving conditions. Specifically, we compute the modified vector field  $\tilde{v}_\theta$  as

$$\begin{aligned} \tilde{v}_\theta(x_t, t; \mathbf{c}) &:= v_\theta(x_t, t; \mathbf{c}_{\{\emptyset, \emptyset\}}), \\ &\quad + w_{\mathbf{a}}[v_\theta(x_t, t; \mathbf{c}_{\{\mathbf{a}, \emptyset\}}) - v_\theta(x_t, t; \mathbf{c}_{\{\emptyset, \emptyset\}})], \\ &\quad + w_{\mathbf{u}}[v_\theta(x_t, t; \mathbf{c}_{\{\emptyset, \mathbf{u}\}}) - v_\theta(x_t, t; \mathbf{c}_{\{\emptyset, \emptyset\}})], \end{aligned} \quad (14)$$

where  $\mathbf{c}_{\{x, y\}}$  denotes a driving condition with the conditions  $x$  and  $y$ .  $\mathbf{a}$  denotes the avatar audio, and  $\mathbf{u} = \{\mathbf{a}_u, \mathbf{m}_u\}$  is the set of user audio  $\mathbf{a}_u$  and user motion  $\mathbf{m}_u$ .  $w_{\mathbf{a}}$  and  $w_{\mathbf{u}}$  are the CFG scales of the avatar audio and user condition, respectively. We use 10% dropout rate for each condition during training.

**Motion Inference Details** We provide more details on our inference strategy in Algorithm 2.

While the lookahead attention enables to generate temporally consistent motion across the blocks, naively introducing it incurs additional latency as it requires  $l$  future

frames. To tackle this problem, we introduce an offset  $\mathbf{O}^i$  for  $i$ -th block generation. Specifically, the offset  $\mathbf{O}^i$  consists of the last  $l$  clean motion latents and the corresponding condition from the previous block:

$$\mathbf{O}^{i+1} = (\mathbf{m}_1^i[-l:], \mathbf{c}^i([-l:])), \quad (15)$$

where  $i = 1, \dots, L - 1$ . These offset motion frames are concatenated with the current noisy motion block  $\mathbf{m}_{t_j}^i \in \mathbb{R}^{B \times d}$  and the condition  $\mathbf{c}^i \in \mathbb{R}^{3 \times B \times d}$  along the time axis, resulting in  $l + B$  motion frames and corresponding conditions (Line 7 in Algorithm 2). Due to the flexibility of diffusion forcing, we can assign difference flow time schedules, i.e.,  $t = 1$  for the offset and  $t = t_j$  for the current noisy latent block.

As we compute the modified vector field  $\tilde{v}_\theta$  for CFG as in Eq. (14), we separately cache and update the KV of these vector fields  $v_\theta(\cdot; \mathbf{c}_{\{\emptyset, \emptyset\}})$ ,  $v_\theta(\cdot; \mathbf{c}_{\{\mathbf{a}, \emptyset\}})$ , and  $v_\theta(\cdot; \mathbf{c}_{\{\emptyset, \mathbf{u}\}})$ . To obtain the KV caches of the clean block, we compute all three by reusing the generated motion block along with the existing KV caches [10]. Moreover, due to the introduction of lookahead attention and offset, the KV caches are updated except the for last  $l$  frames and these frames are provided by the offset. Therefore, the maximum cache size is  $M = L - B - l = 38$ .

## D.2. Training Details

We train the vector field model  $v_\theta$  in Eq. (6) for 2000k steps while freezing the motion latent auto-encoder. We use L1 distance for  $\|\cdot\|$ . For fine-tuning the model using the proposed preference optimization method in Eq. (7), we set the balancing coefficient to  $\lambda = 0.1$  and the deviation parameter to  $\beta = 1000$ . We initialize the reference model  $v_{\text{ref}}$  with the same weights as the trained  $v_\theta$ . We fine-tune  $v_\theta$  for 5k steps and observe that additional tuning does not yield further performance gains.

## D.3. Baseline Implementation

One major challenge of evaluating an interactive head avatar model is the **absence of official implementation of baseline methods**. To bridge this gap, we reproduce INFP [24] on the motion latent space of [11], following its core module, Motion Guider and denote the reproduced model as INFP\*. Based on the description in the original paper, we adopt a bidirectional Transformer encoder for motion generation, where a single window consists of  $N = 75$  frames and additional 10 frames serve as context frames. For Motion Guider, we set  $K = 64$  for both verbal and non-verbal motion memory banks. We train INFP\* on our dataset for 2000k steps.

## D.4. Additional Ablation Studies

**Comparison with Autoregressive Diffusion** We compare our motion latent diffusion forcing with standard au-

toressive diffusion, where the motion generator is conditioned on clean context motion latents. Specifically, we train the motion generator in an autoregressive diffusion manner using four clean context blocks (40 frames) and one noisy block (10 frames). As shown in Fig. 11, autoregressive diffusion suffers from degraded long-horizon generation, whereas diffusion forcing is much more robust to motion drift, highlighting its necessity.

**Ablation on Motion Motion block Size** In Tab. 7, we provide an ablation study on motion block size under a fixed number of training frames  $N = 50$ . Increasing block size (i.e., reducing the number of frames in each block) leads to lower latency while achieving quantitative performance. Conversely, reducing the block size (i.e., increasing the number of frames in each block) can improve the temporal consistency (FVD) and Lip-sync quality (LSE-D) with higher latency.

**Additional Quantitative Results** In Tab. 6, we present the ablation studies on our model with additional metrics, including Visual Quality and Lip Synchronization. We provide a video results of the ablation study. Please refer to the videos “02\_ablation\_wo\_user\_motion.mp4” and “02\_ablation\_wo\_DPO.mp4”. Moreover, we provide video ablation results on the attention mask, where each masking method is illustrated in Fig. 12. The motion jittering observed when using only the blockwise causal mask is clearly visible in the video results, yet difficult to capture with quantitative metrics. We highly recommend watching the ablation video “02\_ablation\_attention\_mask\_XX.mp4”.

## D.5. Evaluation Metrics

**Reactiveness and Motion Richness** Reactiveness and Motion Richness are computed using the EMOCA-based [6] 3D morphable models (3DMMs) that model the facial dynamics via 50-dim expression parameters and 6-dim pose parameters. We extract those parameters using an off-the-shelf 3DMM extractor, SPECTRE [8], for each video frame. Let us denote  $x \in \mathbb{R}^{N \times d}$  as the ground-truth user parameters,  $y \in \mathbb{R}^{N \times d}$  as the ground-truth avatar parameters, and  $\hat{y} \in \mathbb{R}^{N \times d}$  as the generated avatar parameters.  $L$  is the number of frames and  $d$  is the feature dimension. As reported in Tab. 1 and Tab. 4, we can compute rPCC, SID, Var, and FD for expression and pose, respectively.

- **rPCC** (residual Pearson Correlation Coefficients) [17] is to measure the motion synchronization between the user parameters and avatar parameters. Specifically, L1 distance is used to measure the discrepancy between generated PCC and ground-truth PCC where we define PCC as a function of  $z \in \mathbb{R}^{N \times d}$  given ground-truth user



Figure 11. Comparison of **autoregressive diffusion** and **diffusion forcing**. Autoregressive diffusion suffers from motion drift (red arrow) over the long horizon, whereas diffusion forcing maintains stable motion generation over the long horizon.

Table 6. **Ablation study with additional metrics** on user motion and preference optimization. “w/  $\mathbf{m}_u$ ” indicates whether the user motion latent  $\mathbf{m}_u$  is provided as input to the model during both training and inference.

Method		Reactiveness		Motion Richness		Visual Quality			Lip Synchronization	
w/ $\mathbf{m}_u$	DPO	rPCC-Exp ↓	rPCC-Pose ↓	SID ↑	Var ↑	FID ↓	FVD ↓	CSIM ↑	LSE-D ↓	LSE-C ↑
✗	✗	0.052	0.175	2.165	1.586	28.746	185.593	0.818	8.260	6.423
✓	✗	0.042	0.146	2.236	1.408	25.600	175.322	<b>0.854</b>	8.160	<b>6.803</b>
✓	✓	<b>0.003</b>	<b>0.036</b>	<b>2.442</b>	<b>1.734</b>	<b>24.328</b>	<b>170.874</b>	0.833	<b>8.060</b>	6.723

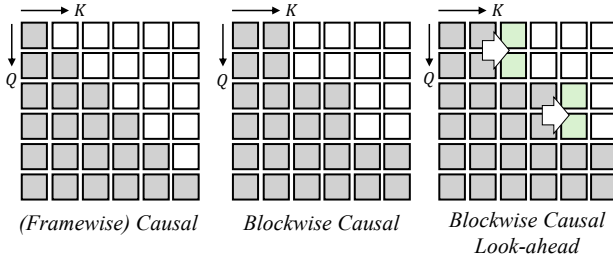


Figure 12. **Attention Mask Comparison.** (Left) frame-wise causal mask; (Middle) block-wise causal mask; (Right) block-wise causal look-ahead mask (Ours).

parameters  $x$ :

$$\text{PCC}(z|x) = \frac{\sum (z_i - \bar{z})(x_i - \bar{x})}{\sqrt{\sum (z_i - \bar{z})^2 \sum (x_i - \bar{x})^2}}, \quad (16)$$

where  $i \in [0, N]$  is the frame index, and  $\bar{z}$  and  $\bar{x}$  denote the mean of  $z$  and  $y$ , respectively. Based on this notation, we can define rPCC as  $|\text{PCC}(y|x) - \text{PCC}(\hat{y}|x)|$ .

- **SID** (Shannon Index for Diversity) [14] is to measure the motion diversity of the generated avatars using  $K$ -means clustering on 3DMM parameters. Following [14], we compute the average entropy (Shannon index) of the clusters with  $K = 15, 9$  for expression and pose, respectively.
- **Var** is the variance of the parameters from generated avatars, which is computed along the time axis and then averaged along the feature axis.
- **FD** (Frechet Distance) [15] measures the distance between the expression and pose distributions of the gen-

Table 7. Ablation studies on motion block sizes on RealTalk.

# blocks	Latency (s) ↓	rPCC-Exp ↓	rPCC-Pose ↓	FVD ↓	SID ↑	LSE-D ↓
10	<b>0.3</b>	0.012	0.056	222.47	<u>2.355</u>	7.290
2	1.5	<b>0.003</b>	<b>0.031</b>	<b>155.81</b>	2.145	<b>6.555</b>
<b>5 (Ours)</b>	0.5	<b>0.003</b>	<u>0.036</u>	<u>170.87</u>	<b>2.442</b>	<u>6.723</u>

erated avatars and the ground truth by calculating

$$|\mu_{\hat{y}} - \mu_y| + \text{tr}(\Sigma_{\hat{y}} + \Sigma_y - 2(\Sigma_{\hat{y}}\Sigma_y)^{\frac{1}{2}}), \quad (17)$$

where  $\mu$  and  $\Sigma$  are the mean and the covariance matrix, respectively.

**Visual Quality** We utilize FID [15] and FVD [18] to assess the image and video quality of the generated avatars, and CSIM [7] to measure the identity preservation performance of avatar generation models.

- **FID** (Frechet Inception Distance) measures the quality of the generated frames by comparing the distribution of image features extracted from a pre-trained feature extractor [15]. The FD computation in Eq. (17) is adopted using the extracted image features.
- **FVD** (Frechet Video Distance) quantifies the spatio-temporal quality of the generated videos by comparing the feature distributions of real and generated videos in a learned video feature space [18]. It reflects both frame-wise quality and temporal consistency. The FD computation in Eq. (17) is adopted using the extracted video features.
- **CSIM** (Cosine Similarity for Identity Embedding) evaluates identity preservation by computing cosine similarity between the facial embeddings from the generated



and the source image, extracted using ArcFace [7].

**Lip Synchronization** We compute LSE-D and LSE-C [4] to assess the alignment between the generated lip motion and the corresponding audio.

- **LSE-D and LSE-C** (Lip Sync Error Distance and Confidence): Both metrics are derived from a pre-trained SyncNet-based audio-visual synchronization model. LSE-D measures the distance between the audio and lip embeddings, where lower values indicate better synchronization. LSE-C measures the confidence score of synchronized audio-visual pairs, where higher values indicate more accurate lip-audio alignment.

## D.6. Human Evaluation

In Fig. 13, we show the interface used for our human evaluation. To improve the evaluation consistency, we additionally provided participants with a reference test and answer sheet. We asked 42 participants to compare 12 videos based on 5 evaluation metrics and indicate their preference. We also provide a video test sheet. Please refer to “04\_human\_evaluation\_XX.mp4”.

## D.7. Supplementary Visual Results

**Comparison with Interactive Head Avatar** We provide the video results to further support the visual results in Fig. 6. Please refer to “01\_interactive\_avatar\_comparison\_XX.mp4”. We also provide a video comparison results using the DEMO videos of Official INFP [24]. Please refer to “01\_interactive\_avatar\_comparison\_demo.mp4”.

**Comparison with Talking Head Avatar** In Fig. 14, we compare our model with SadTalker [22], Hallo3 [5], FLOAT [11], and INFP\* [24] for talking head avatar generation by dropping the user condition at inference. Avatar Forcing can generate competitive results compared to state-of-the-art models, while our model successfully reflects user signals. We also provide the video comparison results. Please refer to “03\_talking\_XX.mp4”.

**Comparison with Listening Head Avatar** In Fig. 15, we compare our model with RLHG [23], L2L [14], DIM [17], and INFP\* [24] for listening head avatar generation. Avatar Forcing can generate competitive results with more expressive facial expression. Please refer to “03\_listening\_XX.mp4” for video results.

## E. Discussion

**Ethical Consideration** Our method can generate more engaging and interactive head-avatar videos, broadening

positive applications such as virtual avatar chat, virtual education, and other communication tools by providing users with a more immersive experience. However, realistic interactive head avatar videos also pose risks of misuse, including identity spoofing or malicious deepfakes. Adding watermarks to generated videos and applying a restricted license can help mitigate these risks. We also encourage the community to use our generated data to train deepfake detection models.

**Limitation and Future Work** Our system focuses on modeling interactive conversations through a head-motion latent space, which enables natural and expressive interactions. This design limits the modeling of richer bodily cues, such as hand gestures, that contribute to more dynamic communication. Moreover, while our model captures user-driven conversational cues via motion latents, certain scenarios may require more explicit controllability, such as directing eye gaze or emphasizing emotional shifts. We believe that incorporating additional user signals, including eye-tracking or emotion-tracking inputs, can address these limitations. Since our framework imposes no architectural constraints on adding new conditions, such signals can be incorporated in future extensions of our system. While diffusion forcing is robust for long-horizon generation, it does not fully address exposure bias. Addressing this issue in the motion latent space remains future work.

!! Please evaluate following user-interactive human avatar generation !!

For each video set (#1 - #5), you will see one user video (real human input) and two generated videos (Avatar A and B), describing the same conversation scenario. Please watch all videos carefully and choose your **preferred video** based on the following criteria.

**1. Reactiveness**  
How well the avatar's motion reflects the **input user's behavior and context**.  
Evaluate how naturally the avatar responds to the user's video. For example, keeping eye contact or reacting appropriately (e.g., smiling together).

**2. Motion Richness**  
Whether the avatar's movements are **expressive** rather than passive or stiff.  
Assess the richness and liveliness of expressions and head movements.

**3. Verbal Alignment (Lip Sync Accuracy)**  
How accurately the **mouth movements** match the speech audio.  
Check whether **lip motions** are well synchronized with the rhythm, timing, and emotion of the voice.

**4. Non-verbal Alignment**  
How naturally the avatar expresses **non-verbal cues** such as eye contact, **nothing, thinking, hesitating, etc.**  
Evaluate how human-like and contextually appropriate these non-verbal reactions appear in response to the user's behavior.

**5. Overall Preference**  
Your **overall impression and preferences** for the videos.  
Choose the video that feels most natural, expressive, and engaging—one that best resembles a real human-to-human interaction.

For a fair comparison, the videos have been randomly shuffled.

Examples for user-interactive human avatar videos

Please carefully watch these two examples for precise evaluation !

**Video #1 Evaluation**

	Avatar A	Avatar B	Tie
Reactiveness	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Motion Richness	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Lip sync	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Non-verbal Alignment	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Overall Preference	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

Reactiveness: Avatar B  
0:21 she laughs after the user laughs

Lip Sync Accuracy: Tie

Motion Richness & Non-verbal Behavior Alignment: Avatar B  
natural head motion and eyebrow movement.

User-Interactive Human Avatar Video #1

**Video #1 Evaluation**

	Avatar A	Avatar B	Tie
Reactiveness	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Motion Richness	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Lip-sync	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Non-verbal Alignment	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Overall Preference	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 13. **Human evaluation interface.** (Left) Instructions for human evaluation; (Middle) A reference sheet for consistent evaluation; (Right) Test and answer sheet.



Figure 14. **Qualitative comparison on talking head avatar generation.**



Figure 15. Qualitative comparison on listening head avatar generation.



## References

- [1] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18392–18402, 2023. [2](#)
- [2] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems*, 2024. [2](#)
- [3] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart- $\alpha$ : Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023. [2](#)
- [4] Joon Son Chung and Andrew Zisserman. Out of time: automated lip sync in the wild. In *Asian Conference on Computer Vision*, 2016. [5](#)
- [5] Jiahao Cui, Hui Li, Yun Zhan, Hanlin Shang, Kaihui Cheng, Yuqi Ma, Shan Mu, Hang Zhou, Jingdong Wang, and Siyu Zhu. Hallo3: Highly dynamic and realistic portrait image animation with video diffusion transformer. In *Conference on Computer Vision and Pattern Recognition*, 2025. [5](#)
- [6] Radek Daněček, Michael J Black, and Timo Bolkart. Emoca: Emotion driven monocular face capture and animation. In *Conference on Computer Vision and Pattern Recognition*, 2022. [3](#)
- [7] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Conference on Computer Vision and Pattern Recognition*, 2019. [4](#), [5](#)
- [8] Panagiotis P Filntisis, George Retsinas, Foivos Paraperas-Papantoniou, Athanasios Katsamanis, Anastasios Roussos, and Petros Maragos. Visual speech-aware perceptual 3d facial expression reconstruction from videos. *arXiv preprint arXiv:2207.11094*, 2022. [3](#)
- [9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. [1](#)
- [10] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems*, 2025. [2](#), [3](#)
- [11] Taekyung Ki, Dongchan Min, and Gyeongsu Chae. Float: Generative motion latent flow matching for audio-driven talking portrait. In *International Conference on Computer Vision*, 2025. [1](#), [3](#), [5](#)
- [12] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. [1](#)
- [13] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. [1](#)
- [14] Evonne Ng, Hanbyul Joo, Liwen Hu, Hao Li, Trevor Darrell, Angjoo Kanazawa, and Shiry Ginossar. Learning to listen: Modeling non-deterministic dyadic facial motion. In *Conference on Computer Vision and Pattern Recognition*, 2022. [4](#), [5](#)
- [15] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch, 2020. Version 0.3.0. [4](#)
- [16] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. [1](#)
- [17] Minh Tran, Di Chang, Maksim Siniukov, and Mohammad Soleymani. Dim: Dyadic interaction modeling for social behavior generation. In *European Conference on Computer Vision*, 2024. [3](#), [5](#)
- [18] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. [4](#)
- [19] Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Conference on Computer Vision and Pattern Recognition*, 2024. [2](#)
- [20] Yaohui Wang, Di Yang, Francois Bremond, and Antitza Dantcheva. Latent image animator: Learning to animate images via latent space navigation. *arXiv preprint arXiv:2203.09043*, 2022. [1](#)
- [21] Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *Conference on Computer Vision and Pattern Recognition*, 2025. [2](#)
- [22] Wenxuan Zhang, Xiaodong Cun, Xuan Wang, Yong Zhang, Xi Shen, Yu Guo, Ying Shan, and Fei Wang. Sadtalker: Learning realistic 3d motion coefficients for stylized audio-driven single image talking face animation. In *Conference on Computer Vision and Pattern Recognition*, 2023. [5](#)
- [23] Mohan Zhou, Yalong Bai, Wei Zhang, Ting Yao, Tiejun Zhao, and Tao Mei. Responsive listening head generation: a benchmark dataset and baseline. In *European Conference on Computer Vision*, 2022. [5](#)
- [24] Yongming Zhu, Longhao Zhang, Zhengkun Rong, Tianshu Hu, Shuang Liang, and Zhipeng Ge. Infp: Audio-driven interactive head generation in dyadic conversations. In *Conference on Computer Vision and Pattern Recognition*, 2025. [3](#), [5](#)