

Archon: A Unified Multimodal Model for Holistic Digital Human Generation

Anonymous CVPR submission

Paper ID

This supplementary material provides additional implementation details, experimental results and a brief discussion on ethics. In Sec. A, we claim that our work adheres to ethical guidelines. In Sec. B, we elaborate on the architectural designs and training protocols for our semantic and animation tokenizers, followed by the detailed specifications of our multimodal task formulation. Sec. C describes the data acquisition and preprocessing pipelines employed for each modality, including description, script, speech, animation, semantic video and video. Finally, Sec. D presents extended qualitative results showcasing multimodal generation and editing capabilities, alongside further quantitative ablation studies on the semantic-guided video diffusion model. We also provide a [project page](#) to demonstrate multimodal generation results including video and audio for better quality judgment.

A. Ethical Considerations

We acknowledge the dual-use nature of high-fidelity avatar generation. While promising for telepresence and content creation, this technology carries risks. We strictly condemn the misuse of our work for harassment or misinformation and emphasize that this research is intended solely for academic purposes. We utilize data in strict adherence to their licenses. We are fully committed to the CVPR Ethics Guidelines, advocating for safeguards like invisible watermarking and continuous bias monitoring to ensure responsible deployment.

B. Implementation Details

B.1. Semantic Video Tokenizer

Model Architecture. Our semantic tokenizer is built upon a 3D convolutional encoder-decoder architecture, designed to map a semantic video to discrete latent codes. The architecture is fully convolutional, consisting of an encoder, a decoder, and look-up free quantizer. We initialize our model’s weights from a pre-trained MAGVIT-v2 [9] checkpoint and introduce a key architectural modification to achieve a higher spatial compression rate suitable for semantic tokenization. Specifically, we add an additional downsampling

operation at the beginning of the encoder. This is implemented as a 3D convolution with a spatial stride of 2, which immediately halves the spatial resolution ($H \times W$) of the input video. Correspondingly, a final upsampling block is added at the end of the decoder to restore the original resolution. This modification doubles the spatial downsampling factor of the feature maps before they are quantized, leading to a more compressed grid of semantic tokens. The detailed architectures of the encoder and decoder are presented in Table A and Table B, respectively.

The encoder processes an input video tensor of shape $T \times H \times W \times C_{in}$. It begins with a $3 \times 3 \times 3$ 3D convolution with a stride of (1, 2, 2), halving the spatial dimensions (H, W) from the outset. This is followed by a series of downsampling stages. Each stage consists of multiple residual blocks (ResBlocks) followed by a downsampling layer, which is a strided 3D convolution that reduces spatial resolution and, in some stages, temporal resolution. The number of feature channels is progressively increased through the encoder. The final stage consists of additional ResBlocks and a $1 \times 1 \times 1$ convolution to project the features into the desired embedding dimension, D_{emb} , before quantization.

The decoder is architecturally symmetric to the encoder. It takes the quantized latent tensor of shape $T' \times H' \times W' \times D_{emb}$ and reconstructs the video to its original dimensions. It begins with a $3 \times 3 \times 3$ convolution and several ResBlocks. Subsequently, a series of upsampling stages, each comprising multiple ResBlocks and an upsampling layer, progressively increase the spatial and temporal resolution while decreasing the number of channels. Upsampling is performed using nearest-neighbor interpolation followed by a $3 \times 3 \times 3$ convolution. To mirror the encoder’s design, the final layer of the decoder is an upsampling block that doubles the spatial resolution, followed by a final convolution to produce the output video with C_{out} channels.

Training. The model is trained as a VQ-GAN [4], fine-tuning from the aforementioned MAGVIT-v2 checkpoint. The training was conducted on 4 TPU v4 platform and took 140 hours to complete. We train on clips at a 128×128 spatial resolution, with a batch size of 8. The training objective is a composite loss function designed to produce high-

Layer	Kernel / Stride	Output Shape	Output Channels
Input	-	$T \times H \times W$	C_{in}
Conv3D (Strided)	$(3, 3, 3) / (1, 2, 2)$	$T \times \frac{H}{2} \times \frac{W}{2}$	F
For $i = 0$ to $N_{blocks} - 1$: (Downsampling Stage i)			
$N_{res} \times$ ResBlock	$(3, 3, 3) / (1, 1, 1)$	$T_i \times H_i \times W_i$	$F \cdot M_i$
Downsample Conv3D	$(4, 4, 4) / (S_{t,i}, 2, 2)$	$T_{i+1} \times H_{i+1} \times W_{i+1}$	$F \cdot M_{i+1}$
$N_{res} \times$ ResBlock	$(3, 3, 3) / (1, 1, 1)$	$T' \times H' \times W'$	$F \cdot M_{last}$
Group Norm + SiLU	-	$T' \times H' \times W'$	$F \cdot M_{last}$
Conv3D (to Embedding)	$(1, 1, 1) / (1, 1, 1)$	$T' \times H' \times W'$	D_{emb}

Table A. The architecture of our 3D CNN Encoder. The input video has dimensions $T \times H \times W \times C_{in}$. F is the base number of filters, M_i are the channel multipliers for each block, N_{res} is the number of residual blocks per stage, and $S_{t,i}$ is the temporal stride for the i -th downsampling layer.

Layer	Kernel / Stride	Output Shape	Output Channels
Input (Quantized Latent)	-	$T' \times H' \times W'$	D_{emb}
Conv3D	$(3, 3, 3) / (1, 1, 1)$	$T' \times H' \times W'$	$F \cdot M_{last}$
$N_{res} \times$ ResBlock	$(3, 3, 3) / (1, 1, 1)$	$T' \times H' \times W'$	$F \cdot M_{last}$
For $i = N_{blocks} - 2$ down to 0: (Upsampling Stage i)			
$N_{res} \times$ ResBlock	$(3, 3, 3) / (1, 1, 1)$	$T_{i+1} \times H_{i+1} \times W_{i+1}$	$F \cdot M_{i+1}$
Upsample + Conv3D	$(3, 3, 3) / (1, 1, 1)$	$T_i \times H_i \times W_i$	$F \cdot M_i$
Group Norm + SiLU	-	$T \times \frac{H}{2} \times \frac{W}{2}$	F
Upsample + Conv3D	$(3, 3, 3) / (1, 1, 1)$	$T \times H \times W$	F
Conv3D (to Output)	$(3, 3, 3) / (1, 1, 1)$	$T \times H \times W$	C_{out}

Table B. The architecture of our 3D CNN Decoder. The input is the latent tensor of shape $T' \times H' \times W' \times D_{emb}$.

080 fidelity reconstructions that also align with ground-truth se-
 081 mantic segmentation maps. The overall loss function \mathcal{L} is
 082 defined as:

$$083 \quad \mathcal{L} = \mathcal{L}_{recon} + \lambda_{adv} \mathcal{L}_{adv} + \mathcal{L}_{commit} + \lambda_{seg} \mathcal{L}_{seg} \quad (1)$$

084 where the components are as follows: \mathcal{L}_{recon} is L2 recon-
 085 struction loss between the original semantic video x and
 086 the reconstructed semantic video \hat{x} . \mathcal{L}_{adv} is the adversarial
 087 loss from a patch-based temporal discriminator that encour-
 088 ages perceptual realism. We set the adversarial loss weight
 089 λ_{adv} to 0.3. \mathcal{L}_{commit} is the commitment loss from the vec-
 090 tor quantization layer, which regularizes the latent embed-
 091 ding space. \mathcal{L}_{seg} is a pixel-wise cross-entropy loss between
 092 the reconstructed output and the ground-truth segmentation
 093 map. The logits for this loss are derived from the negative
 094 squared L2 distance between the generated pixel colors and
 095 a predefined 21-class color palette, sharpened by a temper-
 096 ature $\tau = 10$. This loss component is crucial for learning
 097 a semantic representation, and we set its weight λ_{seg} to 3.0.
 098 The model is trained end-to-end using the Adam optimizer
 099 with a cosine learning rate schedule and a warm-up phase.

B.2. 3DMM Tokenizer

We develop three distinct tokenizers for identity, expres-
 sion, and pose respectively, which are fundamental compo-
 nents of a 3D facial parametric model. Each tokenizer is an
 autoencoder trained using a vector quantization objective
 (VQ-VAE [8]). While they operate on different input fea-
 tures, they share the same underlying network architecture.
 Below, we detail this architecture and the training procedure
 for each tokenizer.

Model Architecture. Our animation tokenizer is a fully
 convolutional autoencoder that operates on 1D temporal se-
 quences of animation parameters. The architecture consists
 of an encoder, a decoder, and a Residual Vector Quantizer
 (RVQ) [5]. The encoder maps the input sequence to a com-
 pressed latent representation, which is then quantized by the
 RVQ. The decoder reconstructs the original sequence from
 the quantized latents. The encoder and decoder are symmet-
 ric and built upon a series of 1D convolutional layers and
 residual blocks (ResBlocks). The specific configuration is
 detailed in Table C. The architecture is consistent across all

three tokenizers, with the only variation being the input and output feature dimensions.

Training. The three tokenizers for identity, expression, and pose are trained independently on their respective data streams. First, each set of animation parameters (expression, identity, and pose) is normalized independently. We compute the mean and standard deviation for each parameter dimension across the entire training dataset. Then, for each sample, we subtract the mean and divide by the standard deviation. This standardizes the distribution of each parameter, which is essential for stable training of the subsequent quantization model. Second, the training objective is to minimize the reconstruction error, regularized by the vector quantization loss. The total loss function is a weighted sum of an L2 reconstruction loss and a VQ commitment loss, as defined in the original VQ-VAE work. All models are trained using the AdamW optimizer with a cosine learning rate schedule, including a warm-up phase over the first 0.5% of training steps. The base learning rate is set to 4×10^{-4} . The training is performed using 128 TPU v2. The specific hyperparameters for each tokenizer are detailed below:

- **Identity Tokenizer:** Trained for 200k steps with a global batch size of 1024. The RVQ consists of 12 codebooks, each with a size of 512. The reconstruction loss weight is 50.0, and the commitment loss weight is 1.0.
- **Expression Tokenizer:** Trained for 200k steps with a global batch size of 1024. The RVQ uses 8 codebooks, each of size 2048. The loss weights are the same as for the identity tokenizer.
- **Pose Tokenizer:** Trained for 100k steps with a larger global batch size of 1024 to stabilize training. The RVQ has 12 codebooks, each with a size of 512. The loss weights are the same as for the identity tokenizer.

B.3. Multimodal Task Formulation and Training

Our primary objective is to train a single, versatile multimodal model capable of holistic avatar generation. This necessitates that the model understands and generates the wide array of modalities that constitute a digital avatar. To this end, we have meticulously designed a comprehensive suite of 72 multimodal tasks. These tasks are structured to teach the model not only to generate individual modalities but also to understand the intricate relationships and dependencies between them, enabling a cohesive and realistic generation of a holistic avatar.

The model is trained on a rich set of modalities, encompassing textual (description, script), acoustic (speech), semantic (semantic), and a hierarchical set of visual modalities (identity, expression, pose, image). Modalities such as identity, image, and description are considered time-invariant. The other modalities can be either from *past* or *current* depending on

if they are the conditions or the predictions in task definition.

Our training tasks are formulated as sequence-to-sequence problems, where the model is given a set of input modalities and is asked to generate a target output modality. The tasks can be categorized as follows:

- **Continuation Tasks:** These tasks involve predicting the *current* state of a modality given its *past* state (e.g., speech (past) \rightarrow speech (current)). This helps the model learn the temporal dynamics of the modality.
- **Cross-Modal Generation Tasks:** The majority of our tasks fall into this category. The model learns to generate a target modality from one or more different source modalities (e.g., speech (current), identity (time-invariant) \rightarrow expression (current)).
- **Chained Generation Tasks:** The tasks are designed to be composable, enabling a chained generation pipeline in our "Thinking in Modality" (e.g., image (time-invariant) \rightarrow identity (time-invariant), then speech (current) \rightarrow expression (current), etc.). Our task suite includes all these intermediate steps to facilitate such chained inference.

This extensive set of 72 tasks, detailed in Table D, ensures that the model is exposed to a vast number of input-output combinations, fostering a deep multimodal understanding and enabling the generation of high-fidelity, holistic avatars.

C. Multimodal Data Details

In this section, we provide a detailed description of the data acquisition and preprocessing pipeline for the all modalities used in our model: description, script, speech, animation, semantic video and image/video. Our pipeline is designed to extract temporally synchronized multimodal data from a large-scale video source, which are essential for training our holistic avatar generation model.

C.1. Description

The description modality provides a rich, structured representation of the person in the video. This modality is designed to capture a holistic set of attributes encompassing appearance, actions, and environmental context. To ensure consistency and comprehensiveness, we employ Gemini 2.5 Pro [1] for the annotation process. A detailed prompt, shown in Figure D, guides the model to analyze a video and return a structured JSON object containing all discernible attributes.

The resulting JSON object is organized into three primary keys: appearance, action, and environment.

- **appearance:** This category captures the subject's physical characteristics. It includes static attributes

Layer	Output Shape	Details
Encoder		
Input	$T \times C_{in}$	C_{in}
Conv1D	$T \times F$	kernel=3, stride=1
Stage 1	$T/2 \times F$	$2 \times \text{ResBlock}(F)$, Downsample
Stage 2	$T/4 \times F$	$2 \times \text{ResBlock}(F)$, Downsample
Stage 3	$T/4 \times F$	$2 \times \text{ResBlock}(F)$
Bottleneck	$T/4 \times F$	$2 \times \text{ResBlock}(F)$
Conv1D	$T/4 \times D$	kernel=1, stride=1
Residual Vector Quantizer		
Quantization	$T/4 \times D$	Residual VQ
Decoder		
Input	$T/4 \times D$	Quantized Latents
Conv1D	$T/4 \times F$	kernel=3, stride=1
Bottleneck	$T/4 \times F$	$2 \times \text{ResBlock}(F)$
Stage 1	$T/2 \times F$	$2 \times \text{ResBlock}(F)$, Upsample
Stage 2	$T \times F$	$2 \times \text{ResBlock}(F)$, Upsample
Stage 3	$T \times F$	$2 \times \text{ResBlock}(F)$
Conv1D	$T \times C_{out}$	kernel=3, stride=1
Output	$T \times C_{out}$	Tanh activation

Table C. Architecture of the Animation Tokenizer. We denote the input sequence length as T and the feature dimension as $C_{in/out}$. The number of residual blocks per stage is set to 2. The base number of channels is $F = 1024$, and the latent dimension is $D = 128$. Downsampling and upsampling operations have a stride of 2.

such as gender, age_group, ethnicity, and body_build, as well as more detailed features like hair_color, hair_style, facial_features, and detailed descriptions of clothing and other physical_attributes.

- **action:** This section details the dynamic aspects of the subject’s performance. It describes the overall activity_type, emotional expression, and nuanced behaviors such as mouth_action, eyebrow_action, head_action, and gaze_direction. It also captures the overall emotion and energy_level conveyed.
- **environment:** This category provides context for the scene, describing the lighting_conditions, background_description, and the general scene_context (e.g., indoor/outdoor).

This structured approach ensures that our model is trained on a consistent and detailed descriptive modality, enabling it to generate holistic and high-fidelity avatars. An example of the generated JSON data is presented in Fig. C.

C.2. Script

The script modality is derived from word-level caption data, which includes precise start timestamps and durations for each word. This allows for fine-grained alignment between the text, speech, and video. When a audio-video clip is extracted, we query the caption data to find all words whose time intervals overlap with the clip’s duration. These se-

lected words are then concatenated in their original order to form the final script that corresponds precisely to the spoken content within that segment. This method ensures that the script is accurately aligned with its corresponding audio and video, which is critical for training our multimodal model.

The aforementioned process applies to our training data where precise word-level annotations are available. For our test sets, specifically CelebV-HQ [12] and HDTF [11], ground-truth captions are not provided. Therefore, we utilize OpenAI’s Whisper model [7] to transcribe the audio into text. This allows us to evaluate our model’s performance on datasets with automatically generated scripts, reflecting a more realistic, in-the-wild scenario.

C.3. Speech

The speech modality is processed to be tightly synchronized with the corresponding video segment. The raw audio track is first resampled to a standard 16 kHz sampling rate. To enhance signal quality and improve model robustness to acoustic variations, we apply a denoising model [10] to 50% of the audio samples, selected at random. The remaining 50% are left unchanged to improve the robustness of model. To align the audio with the video, we extract an audio segment corresponding to the sampled video clip using its timestamps. If the source audio is shorter than the target duration at the sampled point, we pad the segment with silence to ensure a consistent length across all samples.

Output Modality	Input Modality Combinations
script (c)	[desc (t)], [script (p)], [speech (c)], [expr (c)], [semantic (c)]
speech (c)	[desc (t)], [desc (t), script (c)], [script (c)], [script (c), speech (p)], [speech (p)], [script (c), image (t)], [script (c), semantic (c)], [script (c), speech (p), semantic (c)], [id (t), expr (c)], [semantic (c)], [script (c), id (t)]
image (t)	[desc (t)], [speech (c)], [id (t)], [id (t), expr (t), pose (t)], [desc (t), id (t), expr (t), pose (t)]
identity (t)	[desc (t)], [desc (t), script (c), speech (c)], [speech (c)], [image (t)]
expression (c)	[desc (t)], [desc (t), script (c), speech (c), image (t), id (t)], [script (c)], [script (c), speech (c), id (t)], [speech (c)], [speech (c), id (t)], [speech (c), id (t), image (t)], [speech (c), image (t)], [semantic (c)], [expr (p), speech (c)]
pose (c)	[desc (t)], [speech (c)], [speech (c), id (t), expr (c)], [speech (c), expr (c), image (t)], [speech (c), id (t), expr (c), image (t)], [image (t), id (t), expr (c)], [image (t), expr (c)], [id (t), expr (c)], [semantic (c)]
semantic (c)	[desc (t)], [desc (t), script (c)], [desc (t), script (c), speech (c)], [desc (t), script (c), speech (c), id (t), expr (c), pose (c)], [desc (t), image (t)], [desc (t), script (c), speech (c), id (t), expr (c), pose (c), image (t)], [script (c)], [script (c), image (t)], [script (c), speech (c), id (t), expr (c), pose (c), image (t)], [speech (c)], [speech (c), id (t), expr (c), pose (c)], [speech (c), id (t), expr (c), pose (c), image (t)], [speech (c), image (t)], [speech (c), expr (c), pose (c), image (t)], [image (t)], [image (t), id (t), expr (c), pose (c)], [image (t), expr (c), pose (c)], [image (t), expr (c)], [id (t), expr (c), pose (c)], [semantic (p)], [semantic (p), speech (c)], [semantic (p), expr (c), pose (c)]
description (t)	[speech (c)], [image (t), semantic (c), speech (c)], [image (t), semantic (c), speech (c), script (c)], [image (t)], [image (t), semantic (c)], [id (t), expr (c), pose (c)]

Table D. Overview of the 72 Multimodal Training Tasks. The model is trained to predict the output modality from various input combinations. The state of each modality is indicated in parentheses: (c) for current, (p) for past, and (t) for time-invariant. For brevity in long combinations, we use abbreviations: desc (description), id (identity), and expr (expression).

C.4. Animation

The animation modality in our model is derived from 3D Morphable Model (3DMM) parameters [3], which provide a rich, structured representation of the human head’s geometry and dynamics. These parameters are not directly available in the raw video data and are obtained through a pre-processing step where a 3DMM is fitted to each video frame using an optimization-based approach [2]. This fitting process yields a set of continuous parameters for each frame, which contains facial expression, person-specific identity, head rotation, and head translation.

To make these continuous parameters suitable for our

multimodal language model, we convert them into a sequence of discrete tokens. This tokenization process is crucial as it allows our model to handle animation in a manner analogous to text. The process involves two main steps: normalization and vector quantization. First, we use pre-calculated mean and standard deviation of each parameter to normalize itself independently. Second, the normalized continuous parameters are quantized into discrete tokens using three separate, learned Vector Quantized-Variational Autoencoders [8] (VQ-VAEs). We use distinct VQ-VAEs for expression, identity, and pose (where pose is the concatenation of rotation and translation parameters). Each

Label Name	Label Name	Label Name
background	hair	pupil
face	brows	iris
neck	lashes	sclera
ears	beard	clothes
upper_lip	teeth	glasses
lower_lip	tongue	headwear
nostrils	other_in_the_mouth	accessory

Table E. The 21 semantic labels used for generating semantic segmentation maps.

VQ-VAE has its own learned codebook, effectively creating a unique vocabulary for each animation component. This process maps the high-dimensional, continuous animation signals into compact sequences of discrete tokens. By tokenizing the animation parameters, we can seamlessly integrate them into our transformer-based MLLM architecture, enabling it to learn cross-modal associations between animation, script, speech, and video frames.

C.5. Semantic Video

For the semantic video modality, we process each frame to generate a semantic segmentation map. This provides our model with a context-efficient representation of video, particularly focusing on the human subject. We utilize the DINOv2 model [6] to parse segmentation, which delineates 21 distinct classes corresponding to key facial features, hair, and accessories. This fine-grained labeling scheme enables our model to learn detailed representations of the human head and upper body. The complete list of the 21 semantic labels is provided in Table E. This detailed decomposition is instrumental for the model to generate realistic textures and geometry for each part of the avatar, from the subtle nuances of eye components to various accessories.

C.6. Image/Video

Our video processing pipeline is designed to generate normalized, head-centric video clips. Initially, raw videos are decoded and uniformly downsampled to a frame rate of 30 frames per second (fps). To achieve a consistent, head-centric view, we first detect the facial landmarks to the face in each frame. From the 3DMM fit, we derive a bounding box that encompasses the face and a portion of the upper body. To ensure a stable view without jitter, we compute a common crop region based on the union of these bounding boxes across the entire video sequence. The size of this crop is randomly scaled to introduce variations in framing, from tight close-ups to wider shoulder-level shots. This cropped region is then resized to a final resolution of 256×256 pixels using bicubic interpolation with anti-aliasing.

Methods/Metrics	FID↓	FVD↓
Ours w/o text cond	17.2310	32.3772
Ours w/o joint crossattention	22.7003	36.1064
Ours	16.8678	26.6860

Table F. We show the ablation study on the different condition ways to the diffusion model.

D. More Results

D.1. Multimodal Generation

We present additional qualitative results of our multimodal generation framework. As illustrated in Fig. A (a), when conditioned on a description of a subject (e.g., a man in a suit) and a corresponding script, our model successfully synthesizes all complementary modalities, including speech, animation, semantic segmentation, and the final RGB video. These generated modalities exhibit precise temporal synchronization, with the output video demonstrating high visual fidelity and strict alignment with the input conditions. Besides, in Fig. A (b), we demonstrate generation from semantic inputs; given a semantic video, our model synthesizes a photorealistic video that faithfully adheres to the semantic appearance and motion. Furthermore, the model demonstrates its understanding capabilities by accurately inferring the description, script, speech, and animation directly from the semantic video input.

D.2. Modality-specific Editing

We further demonstrate our model’s versatility in multimodal editing in Fig. B. In Fig. B (a), we perform script editing, where we synthesize a new talking video that articulates a modified script while faithfully preserving the subject’s original appearance and vocal identity. In Fig. B (b), we illustrate disentangled attribute editing by altering the speaker’s gender from male to female while retaining fine-grained attributes such as hairstyle and clothing. Notably, the model simultaneously adapts the generated voice to align with the modified visual appearance, demonstrating robust cross-modal consistency. Finally, Fig. B (c) depicts animation editing (face reenactment); by utilizing 3DMM coefficients extracted from a reference video, we drive the source subject to replicate the reference’s pose and expressions with high fidelity.

D.3. Ablation of Semantic-driven Video Diffusion Model

We investigate the impact of different conditioning mechanisms on the video diffusion backbone, with quantitative results summarized in Tab. F. All ablations are performed on the held-out test split. Given the dense temporal nature of the semantic video input, we employ channel-wise con-

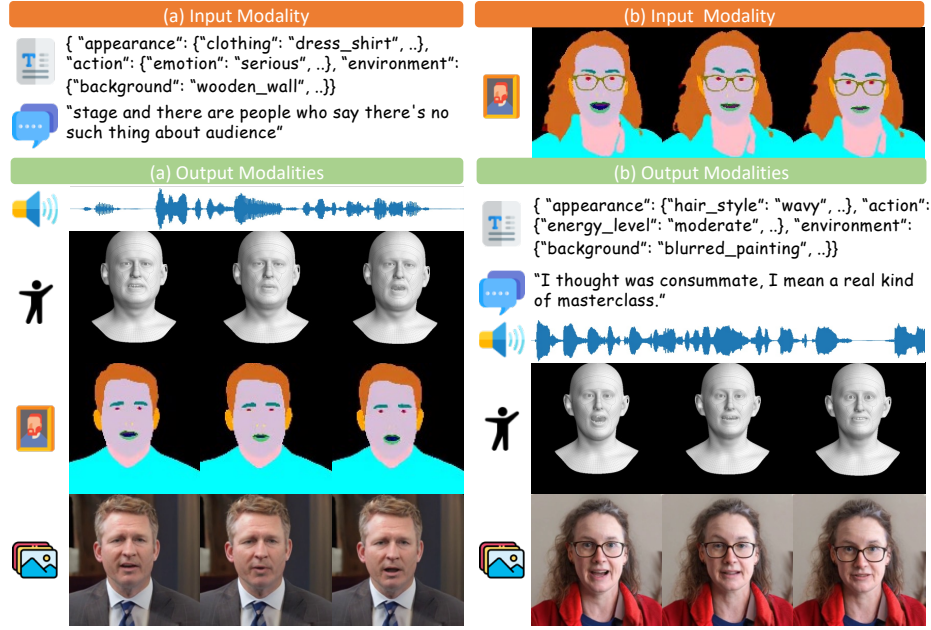


Figure A. **Multimodal Generation.** We show our model is capable of doing text or speech-to-all generation and can support a variety of cross-modality reasoning and generation tasks.

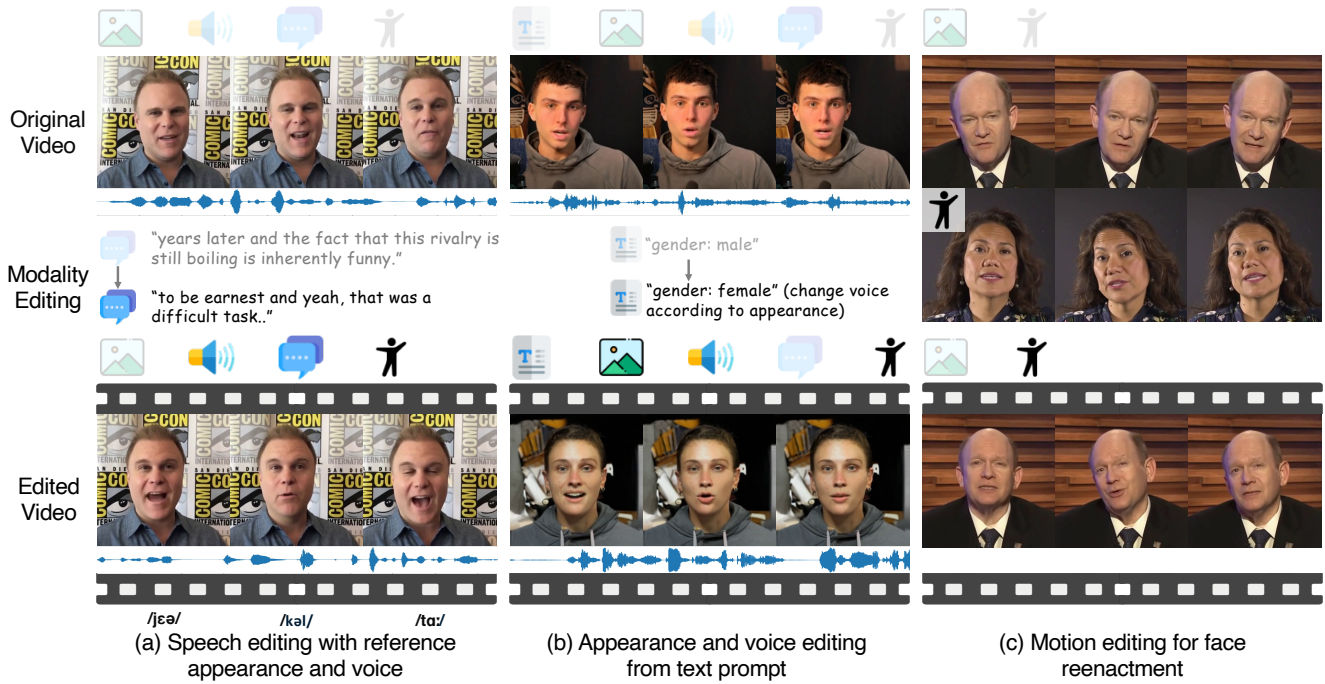


Figure B. **Modality-specific Editing.** We demonstrate that our model support editing mutlimodal input with remarkable flexibility - modifying arbitrary chosen modal while maintains the others untouched. The icons on the top showing the modalities used in the example, and the highlighted icons are the ones that are varied.

catenation with the noisy latents rather than cross-attention, adhering to computational memory constraints. In Sec. 3.3, we propose concatenating the reference image with its corresponding segmentation mask for cross-attention ("Ours").

This strategy establishes a structural bridge, facilitating the effective transfer of appearance from the reference to the synthesized video. We contrast this with a baseline that utilizes only the RGB reference image in the cross-attention

module (“Ours w/o joint cross-attention”). As observed in Tab. F, the degradation in FID and FVD scores in the absence of segmentation image underscores their critical role in guiding the diffusion model to accurately map reference appearance to generated motion. Finally, we assess the contribution of textual prompts. The configuration “Ours w/o text cond” omits text embeddings from the cross-attention layers. The resulting decline in metrics confirms that multi-modal conditioning is essential for maintaining high video fidelity.

References

- [1] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Naveen Sachdeva, Inderjit Dhillon, Marcel Blisstein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 3
- [2] Radek Daněček, Michael J Black, and Timo Bolkart. Emoca: Emotion driven monocular face capture and animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20311–20322, 2022. 5
- [3] Bernhard Egger, William AP Smith, Ayush Tewari, Stefanie Wuhler, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, et al. 3d morphable face models—past, present, and future. *ACM Transactions on Graphics (ToG)*, 39(5):1–38, 2020. 5
- [4] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 1
- [5] Robert Gray. Vector quantization. *IEEE Assp Magazine*, 1(2):4–29, 1984. 2
- [6] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 6
- [7] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023. 4
- [8] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 2, 5
- [9] Lijun Yu, José Lezama, Nitesh Bharadwaj Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, et al. Language model beats diffusion-tokenizer is key to visual generation. In *ICLR*, 2024. 1
- [10] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2021. 4
- [11] Zhimeng Zhang, Lincheng Li, Yu Ding, and Changjie Fan. Flow-guided one-shot talking face generation with a high-resolution audio-visual dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3661–3670, 2021. 4
- [12] Hao Zhu, Wayne Wu, Wentao Zhu, Liming Jiang, Siwei Tang, Li Zhang, Ziwei Liu, and Chen Change Loy. Celebv-hq: A large-scale video facial attributes dataset. In *European conference on computer vision*, pages 650–667. Springer, 2022. 4


```

{
  "appearance": {
    "gender": "male",
    "age_group": "adult",
    "ethnicity": ["caucasian"],
    "body_build": ["average"],
    "hair_color": ["black"],
    "hair_style": ["short"],
    "facial_features": ["none_discernible"],
    "clothing": {
      "upper_body": "button-down_shirt_over_t-shirt",
      "lower_body": "none",
      "footwear": "none",
      "accessories": ["watch", "ring", "bracelet"],
      "dominant_colors": ["black", "blue", "white", "red"]
    },
    "physical_attributes": {
      "visible_tattoos": "none",
      "visible_piercings": "none",
      "distinctive_marks": "none",
      "posture": "upright",
      "gait": "not_applicable",
      "physical_aids": ["none"]
    }
  },
  "action": {
    "activity_type": "speaking",
    "expression": "smile",
    "overall_impression": ["gesturing_with_hands_while_speaking", "raising_hands_to_chest_level", "making_a_fist"],
    "emotion": ["positive_and_engaging", "enthusiastic"],
    "energy_level": ["medium_to_high", "animated"],
    "mouth_action": ["wide_opening", "narrow_opening", "relaxed_lips", "pulling_of_lip_corners", "synchronization_and_precision_with_spoken_words"],
    "eyebrow_action": ["raising_inner_eyebrows", "raising_outer_eyebrows", "brow_furrowing"],
    "blink_frequency": "medium",
    "head_action": ["mostly_centered", "tilts_inquisitively", "nods_rhythmically", "amplitude_and_tempo_of_movements", "directional_changes_pitch/yaw/roll", "frequency_of_changes", "transitions_between_stillness_and_motion"],
    "eye_state": ["fully_wide_alert"],
    "gaze_direction": ["straight_ahead", "fixed_forward_focus"],
    "nonverbal_habits": ["habitual_eyebrow_flicks", "frequent_micro-nods", "timing_and_context_of_cues", "consistency_of_cues", "typical_amplitude_subtle_vs_pronounced"],
    "interactions": ["with_object_e.g._microphone/instrument", "none"],
    "props_used": ["microphone", "none"]
  },
  "environment": {
    "lighting_conditions": "bright",
    "background_description": "light_colored_wall_with_a_framed_certificate_and_a_model_of_a_watch_on_a_shelf",
    "time_of_day": "unknown",
    "weather_conditions": "indoor_not_applicable",
    "context": "indoor"
  }
}

```

Figure C. An example of the structured JSON description obtained from Gemini 2.5 Pro.

Enhanced Video/Image Analysis Prompt

You are an expert video and image analysis AI. Your task is to meticulously analyze a given video or sequence of images featuring a person performing, and extract all possible relevant attributes about the person, their actions, and the surrounding environment.

Critical Instruction for JSON Output

Your output MUST be a single, complete, and perfectly valid JSON object. Do NOT include any introductory or concluding text, explanations, or extraneous characters outside of the JSON. Ensure proper syntax, including all commas, brackets, and quotes.

Data Extraction Schema

Here's the information you need to extract, grouped into three main categories, and the precise JSON format you must adhere to. For attributes specified as lists (e.g., body_build, hair_color, hair_style, facial_features, accessories, dominant_colors, physical_aids, interactions, props_used, overall_impression, Mouth_Action, Eyebrow_Action, Head_Action, Eye_State, Gaze_Direction, Nonverbal_Habits), ensure you return a list containing ALL discernible attributes. The attribute values given below are just examples, but use them for inspiration. Attributes should be strings with underscores instead of space where necessary.

```
{
  "appearance": {
    "gender": "male/female/non-binary/unknown",
    "age_group": "child/preteen/teenager/young_adult/adult/older_adult/senior/unknown",
    "ethnicity": ["list_ethnicities_if_discernible", "caucasian", "asian", "african_american", "hispanic", "middle_eastern", "unknown", "etc."],
    "body_build": ["slim", "average", "athletic", "muscular", "heavy", "unknown"],
    "hair_color": ["black", "brown", "blonde", "red", "gray", "white", "green", "blue", "pink", "other", "unknown"],
    "hair_style": ["bald", "long", "short", "curly", "straight", "wavy", "..."],
    "eye_color": ["light_brown", "dark_brown", "light_blue", "..."],
    "eye_style": ["long_eyelashes", "short_eyelashes", "..."],
    "teeth": ["straight", "missing_teeth", "..."],
    "facial_features": ["glasses", "beard", "wrinkly_skin", "elastic_skin", "clean_skin", "sideburns", "mustache", "freckles", "..."],
    "clothing": {
      "upper_body": ["description_of_upper_garment_e.g._t-shirt/dress_shirt/blouse/hoodie/jacket/sweater/tank_top/sports_bra/suit/tie/none"],
      "lower_body": ["description_of_lower_garment_e.g._jeans/trousers/skirt/shorts/leggings/sweatpants/none"],
    },
    "physical_attributes": {
      "visible_tattoos": "description_of_tattoos_and_location_if_visible_e.g._full_sleeve_right_arm/small_design_neck/none",
    },
  },
  "action": {
    "activity_type": "dancing/singing/playing_instrument/speaking/acting/sports/walking/running/sitting/standing/other",
    "expression": "smile/frown/closed_mouth/etc.",
    "overall_impression": ["brief_descriptive_summary_of_the_performance", "e.g._head_turn_to_the_left", "e.g._raising_her_left_arm"],
    "emotion": ["description_of_general_energy_and_dominant_emotional_tone", "e.g._animated_and_joyful", "..."],
    "energy_level": ["overall_magnitude_of_emotional_display", "e.g._faint_micro-expressions_to_vivid_reactions", "e.g._consistently_low-key"],
  },
  "environment": {
    "lighting_conditions": "bright/dim/natural_light/artificial_light/backlit/etc.",
    "background_description": "detailed_description_of_the_background_e.g._blurred_trees/crowded_street/empty_white_wall",
    "time_of_day": "morning/afternoon/evening/night/unknown",
    "weather_conditions": "sunny/cloudy/rainy/snowy/indoor_not_applicable/etc.",
    "scene_context": "indoor/outdoor/stage/street/home/office/natural_environment/etc."
  }
}
```

Figure D. The part of prompt provided to Gemini 2.5 Pro for video annotation.