DIRECTBEAT: BEHAVIORAL PLANNING AND RHYTHMIC EXECUTION FOR CO-SPEECH MOTION

Anonymous authors
Paper under double-blind review

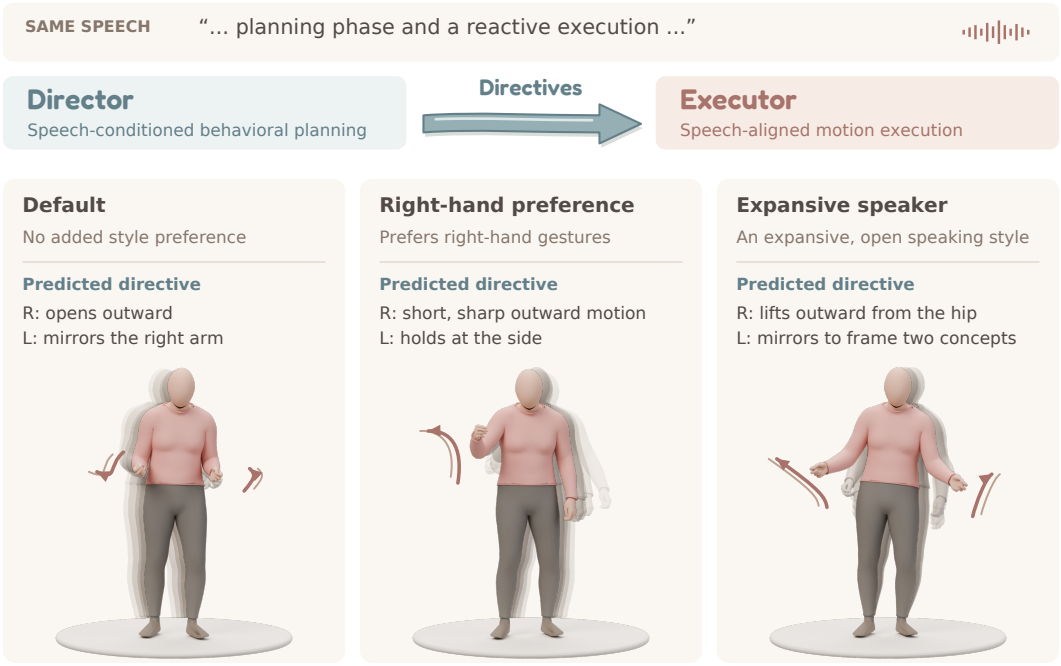

Figure 1: **DirectBeat: from behavioral plans to body motion.** The Director predicts explicit directives, and the Executor generates motion aligned with speech. System prompt edits change gesture preferences while speech and the executor remain fixed.

ABSTRACT

Speech-to-motion correspondence is highly ambiguous, making generation difficult to control and inspect. We present *DirectBeat*, which connects behavioral planning and motion execution through explicit directives. Using pretrained knowledge, speech context, recent memory, and retrieved examples, a frozen multimodal language model predicts two-second behavioral plans. A specialized executor realizes these plans as motion. Warp-VQVAE learns audio-conditioned latent space re-timing via temporal warping to synchronize motion with speech. BEAT2-Directive supplies 101,660 aligned speech–directive–motion triplets from BEAT2 to supervise this interface. DirectBeat achieves FGD scores of 3.631 and 2.76 on the single- and all-speaker benchmarks, reducing FGD by 11% and 38% compared with prior methods. On the unseen TED speech benchmark, it receives the highest aggregate human preference across realism, semantic consistency, synchrony, and diversity among the compared methods. We further demonstrate an application that directives support inspection and persona editing in language without retraining the executor, providing a reusable motion interface for LLM agents. Video results are provided in the supplementary material.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Speech-to-motion generation is inherently ambiguous: the same utterance can accompany pointing, emphasis, illustrative gestures, or pauses (De Ruiter et al., 2012; Burgoon et al., 1990; McNeill, 1992). Paired recordings capture individual realizations of this one-to-many mapping (Alexanderson et al., 2020; Habibie et al., 2021). Their limited scale and contextual coverage leave behavioral choices weakly specified. Learning gesture selection and motion dynamics from such supervision makes direct generation difficult to control and inspect.

Direct speech-conditioned generators learn this mapping within a motion model (Liu et al., 2024; Ao et al., 2022; Pang et al., 2023; Yi et al., 2023; Chen et al., 2024b). Semantic retrieval, intention reasoning, and prompt-conditioned methods provide additional guidance (Zhang et al., 2024; 2025; Liu et al., 2025a; Chen et al., 2024a). We address the underspecified behavioral choice through an explicit *directive interface*. Before generating motion, the system describes the intended action, posture, and phase in language. This gives the motion model a concrete target and lets users inspect or edit behavior before execution.

We introduce **DirectBeat**, a dual-process[1] system with a language-model director and a learned motion executor. In the *planning* phase, a frozen multimodal language model combines knowledge from large-scale pretraining, recent directive memory, corpus priors, and retrieved motion examples. These inputs guide the choice of an actionable plan among plausible gestures, using knowledge beyond the paired motion corpus to predict behavior. Persona preferences can be customized by system prompting the director without retraining the executor (Figure 1).

In the *execution* phase, the executor maps explicit action descriptions to motion tokens. **BEAT2-Directive** adds motion-grounded descriptions to BEAT2 (Liu et al., 2024; 2022b), providing supervision in the director's schema of motion phase, gesture class, posture, and bilateral actions. Two-second language plans leave framewise trajectories and acoustic alignment unspecified. The motion model supplies these details through timed transcript conditioning and **Warp-VQVAE**, which learns latent space temporal warping for adpative re-timing. Directives therefore control what the avatar does, while the executor and decoder determine how it moves and synchronizes with speech.

We evaluate both the complete system and the roles of planning and execution. DirectBeat achieves FGD scores of 3.631 and 2.76 on the single- and all-speaker BEAT2 benchmarks and receives the highest aggregate preference among the compared methods on unseen TED speech. With the executor fixed, predicted directives reduce FGD from 3.920 to 3.631. Retrieval, component ablations, and director replacements further examine the directive representation and its use in generation. The supplementary videos show how prompt edits change predicted directives and animated performances.

Our contributions are:

- **DirectBeat**, a co-speech motion system that combines pretrained language-model knowledge and contextual planning with actionable directives and learned rhythmic motion execution, which can produce high quality co-speech motion on BEAT2 benchmark, and achieves more aggregate preference on the unseen TED speech benchmark compared with prior methods.
- **Warp-VQVAE**, a motion tokenizer with learned audio-conditioned temporal warping for latent space re-timing, which allows coordinate discrete behavioral segments with acoustic timing, enhancing LLM behavior planning with life-like rhythmic motion.
- **BEAT2-Directive**, an extension of BEAT2 with 101,660 aligned speech–directive–motion triplets and motion-grounded, human-filtered annotations for directive supervision.
- We further show an application that our system with directives interface can support inspection and persona editing in language without retraining the executor on unseen public speech, providing a reusable motion module for LLM agents.

[1] "Dual process" refers to behavioral planning and motion execution, inspired by the distinction between deliberation and execution (Kahneman, 2011).

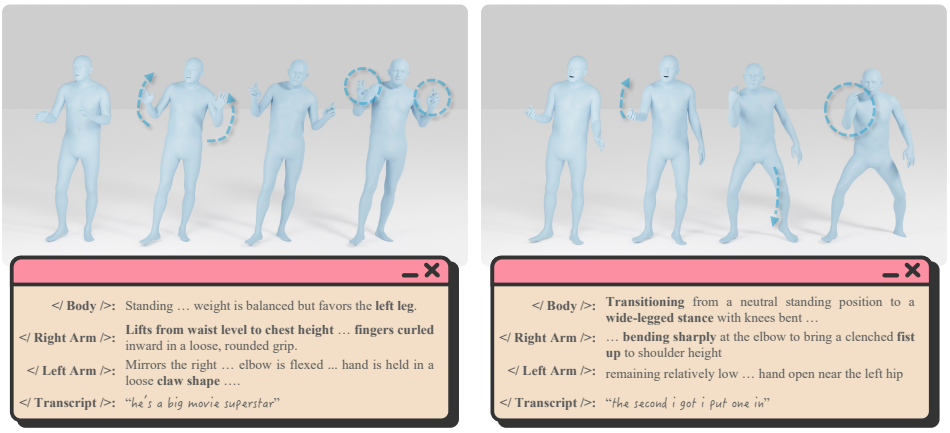

Figure 2: **BEAT2-Directive supplies aligned supervision for execution.** Motion-derived descriptions specify bilateral actions and body posture. The director uses the same physical action fields when predicting a plan from speech.

2 Related Work

Co-speech generation and semantic planning. Speech-conditioned models use acoustic and linguistic features to synthesize motion (Liu et al., 2024; Yi et al., 2023; Chen et al., 2024b; Ao et al., 2022). Semantic methods incorporate gesture retrieval or semantic guidance (Zhang et al., 2024; 2025; Cheng et al., 2024), while intention-conditioned work combines language-model reasoning with motion generation (Liu et al., 2025a). DirectBeat builds on these approaches by predicting a sequence of body-specific behavioral plans and executing them with a model trained on aligned directives and a learned acoustic timing mechanism.

Prompt-conditioned generation and motion descriptions. GestureDiffuCLIP and FreeTalker demonstrate language-conditioned motion control (Ao et al., 2023; Yang et al., 2024). SynTalker and CoordSpeaker combine speech with supplied motion descriptions (Chen et al., 2024a; Fang et al., 2026). DirectBeat connects language control to behavioral planning: the director predicts sequential, body-specific directives from speech and recent context, and the executor realizes these editable plans with transcript conditioning and audio-conditioned temporal warping. Appendix A further discusses motion descriptions and representations.

Rhythm-aware motion representations. Rhythmic Gesticulator (Ao et al., 2022) normalizes beat-defined motion blocks and learns a VQ gesture lexicon, combining rhythm normalization with a quantized representation. Warp-VQVAE instead learns audio-conditioned resampling fields within the latent autoencoder, before quantization and before motion decoding. This design incorporates acoustic alignment into both the learned motion representation and its reconstruction.

3 DirectBeat

An overview of DirectBeat is shown in Figure 3. Given audio A and a word-aligned transcript W, DirectBeat connects its director and executor through a *directive interface* (Section 3.1). Directives use the LLM's native language representation to make planned actions inspectable and editable. The director (Section 3.2) draws on pretrained knowledge, recent plans H_t, a corpus summary $\mathcal{K}$, and retrieved examples $\mathcal{E}_t$ to predict plausible motions for successive two-second windows. Its memory of up to two preceding directives provides context for gesture continuations and transitions across windows. The executor (Section 3.3) maps directives $D = (D_t)_t$ and the transcript to motion tokens $\mathbf{z}$, and Warp-VQVAE (Section 3.4) decodes them with audio conditioning. This learned execution supplies the fine-grained motion and rhythmic alignment that two-second language plans leave unspecified:

$$D_t \sim p_{\text{dir}}(\cdot \mid A_t, W_t, H_t, \mathcal{K}, \mathcal{E}_t), \quad \mathbf{z} \sim p_\theta(\cdot \mid D, W), \quad \hat{M} = \text{Decode}_\psi(\mathbf{z}, A). \tag{1}$$

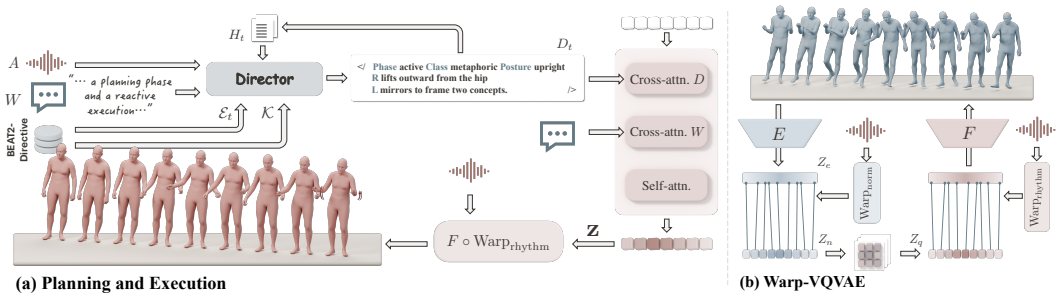

Figure 3: **DirectBeat overview.** (a) For each two-second window, the frozen Director predicts a directive D_t from speech, recent plans H_t, corpus priors $\mathcal{K}$, and retrieved BEAT2-Directive examples $\mathcal{E}_t$. The Executor generates motion tokens $\mathbf{z}$ from directives and time-synchronized transcripts, and the Warp-VQVAE decoder converts them to motion with audio. (b) Warp-VQVAE encodes motion M, adjusts latent timing with the audio-conditioned warp $\mathrm{Warp}_{\mathrm{norm}}$ before quantization Q, and re-times the quantized latents with $\mathrm{Warp}_{\mathrm{rhythm}}$ before decoding. Codebook lookup is implicit in (a).

The director remains frozen; θ and ψ parameterize the executor and Warp-VQVAE, respectively, trained on BEAT2-Directive.

3.1 Directive Interface

Each directive specifies a gesture class, motion phase, body posture, and separate left- and right-arm actions. Descriptions use anatomical landmarks, such as moving a hand toward a shoulder, to express movement in language. One directive per two-second interval provides a sequence of local behavioral goals for the executor. Section 4.2 evaluates this interface through motion retrieval.

BEAT2-Directive Triplet Dataset. We extend BEAT2 with 101,660 aligned `<speech, directives, motion>` triplets. The source corpus contains approximately 60 hours; the annotated two-second windows cover about 56.5 hours. A vision-language model (Comanici et al., 2025) describes the rendered motion without access to speech or transcripts. Human filtering checks that the descriptions match the motion, follow the schema, and distinguish left- and right-arm actions correctly. We then attach the transcripts. Training descriptions are therefore grounded in observed motion, whereas the director predicts plans from speech at inference. Figure 2 shows examples, and Appendix B describes the annotation procedure.

3.2 Director: Speech-Conditioned Behavioral Planning

The multimodal language-model director (Comanici et al., 2025) predicts a structured directive from each speech interval using recent context, corpus knowledge, and a fixed planning procedure.

Memory. The director retains up to two preceding directives, including motion phase, class, posture, and bilateral actions. This history helps it decide whether to continue, hold, retract, or initiate a movement while keeping the prompt length bounded.

Knowledge and retrieval. A compact document summarizes motion-state and gesture-class priors, transition statistics, arm-use asymmetry, and lexical cues from the BEAT2-Directive training split. For each window, we retrieve three transition-oriented and four semantic examples using IDF-weighted transcript overlap and compatibility with the preceding state and class, then remove duplicates. These examples provide concrete action descriptions and transition patterns. The director also supports persona customization through system prompts (Figure 1). Changing the system prompt can alter requested arm-use habits without retraining the executor; the supplementary video shows the corresponding prompt, directive, and motion changes.

Planning. The prompt first requests the window's motion state and then its gesture class, posture, and bilateral actions under a constrained schema. We use `gemini-3.8-flash` for the main configuration, with decoding temperature 0.15. The model weights remain frozen. Section 4.2

reports director ablations and sensitivity analysis; Appendix K gives the planning procedure and deployment details.

3.3 EXECUTOR: LEARNED MOTION EXECUTION

The executor maps planned directives to motion using a conditional masked-token generator (Guo et al., 2024). Separate text encoders encode the text in directives D and transcript W, which are fed into the generator through alternating cross-attention layers. Directive attention specifies the action, while transcript attention supplies lexical and temporal context. Independent condition dropout reduces reliance on any one input. Appendix E describes the architecture and training procedure.

Temporally Synchronized Positional Embeddings. Transcript token positions do not directly correspond to motion time. We align the two streams by mapping BERT subword tokens (Sanh et al., 2019) to the motion-token timeline using TextGrid word timestamps (Boersma, 2001). Shared sinusoidal embeddings place transcript attention and motion tokens in the same temporal coordinate system. Section 4.3 ablates this synchronization; Appendix E details the alignment and conditioning architecture.

3.4 WARP-VQVAE: RHYTHM-AWARE MOTION TOKENIZATION

Directives specify what happens in each two-second window, and the executor predicts tokens on a fixed temporal grid. Warp-VQVAE supplies the finer speech timing by separating *what* a movement is, stored in discrete tokens, from *when* it occurs, predicted from audio. One learned temporal warp adjusts timing before quantization; a second re-times motion from audio before decoding.

Warp operator. Each warp predicts where to sample a latent sequence in time and interpolates it there. For latent features Z over T positions and audio features A aligned to them,

$$G = G_{\text{base}} + \frac{\Delta_{\max}}{T}\tanh\big(\Phi(A, Z)\big), \qquad \text{Warp}(Z, A) = \text{Interp}(Z, G). \tag{2}$$

The uniform grid G_{base} spans $[-1, 1]$, G gives the predicted sampling positions, and $\Delta_{\max}/T$ bounds each displacement. $\text{Warp}_{\text{norm}}$ and $\text{Warp}_{\text{rhythm}}$ share this form but use separate 1D convolutional predictors Φ_{norm} and Φ_{rhythm}, learned jointly with the autoencoder.

Autoencoder. For motion M, encoder E, residual quantizer Q, and decoder F,

$$\begin{aligned} Z_e &= E(M), \quad Z_n = \text{Warp}_{\text{norm}}(Z_e, A), \\ (\mathbf{z}, Z_q) &= Q(Z_n), \quad \hat{M} = F\big(\text{Warp}_{\text{rhythm}}(Z_q, A)\big), \end{aligned} \tag{3}$$

where $\mathbf{z}$ are discrete token indices and $Z_q = \mathcal{C}(\mathbf{z})$ sums their residual codebook embeddings.

Training. Before $\text{Warp}_{\text{rhythm}}$, we randomly jitter Z_q in time by interpolating it on a perturbed grid, $\text{Interp}(Z_q, G_{\text{base}} + \epsilon)$. Because timing retained in the tokens is corrupted, reconstruction encourages $\text{Warp}_{\text{rhythm}}$ to recover timing from audio and the codebook to represent motion structure (Appendix F). We train end-to-end with the standard VQ-VAE objective (Van Den Oord et al., 2017), $\mathcal{L}_{\text{VQ}} = \mathcal{L}_{\text{recon}} + \lambda\mathcal{L}_{\text{commit}}$ with $\lambda = 0.1$.

Generation. The executor supplies tokens $\mathbf{z}$, so only the rhythm path is used: $\text{Decode}_\psi(\mathbf{z}, A) = F\big(\text{Warp}_{\text{rhythm}}(\mathcal{C}(\mathbf{z}), A)\big)$. The director plans at two-second resolution, the executor selects motion content, and the decoder times it to speech.

4 EXPERIMENTS

We evaluate full-body and hand motion on BEAT2 (Liu et al., 2024) under single- and all-speaker protocols. The single-speaker split contains 95 training, 8 validation, and 15 test recordings. We report Fréchet Gesture Distance (FGD) (Yoon et al., 2020), Beat Constancy (BC) (Li et al., 2021b), and L1 diversity (Li et al., 2021a) to measure motion-distribution similarity, speech–motion beat alignment, and motion amplitude, respectively. Our evaluation focuses on body and hand motion; facial animation is outside its scope. Appendices H and M.1 describe the metrics and evaluation settings.

Table 1: **Quantitative comparison on BEAT2.** Results are grouped by speaker protocol and method scope. Bold and underline identify the best and second-best FGD and BC within each speaker protocol. L1 diversity measures motion amplitude; APD and Coverage@5 measure feature dispersion and real-motion coverage.

(a) Single speaker

Method	FGD ↓	BC ↑	L1 div.
Holistic motion			
DiffSHEG	8.986	0.714	11.91
TalkShow	6.209	0.695	13.47
ProbTalk	5.040	0.771	13.27
EMAGE	5.512	0.772	13.06
MambaTalk	5.366	**0.781**	13.05
SemTalk	4.278	<u>0.777</u>	12.91
Full-body motion			
CaMN	6.644	0.676	10.86
HA2G	12.14	0.671	8.916
DisCo	9.417	0.643	9.912
SynTalker	4.687	0.736	12.43
GestureLSM	<u>4.088</u>	0.714	13.24
Ours	**3.631**	**0.781**	10.51

(b) All speakers

Method	FGD ↓	BC ↑	L1 div.
Full-body motion			
CaMN	5.12	0.200	5.58
Audio2Photoreal	8.49	0.326	6.24
ReMoDiffuse	11.20	0.218	5.06
DSG	11.74	0.734	11.12
HoloGest	6.46	**0.803**	13.53
EMAGE	6.92	0.284	6.06
RAG-GESTURE	4.87	0.514	9.94
GestureLSM	<u>4.46</u>	0.525	9.23
Ours	**2.76**	<u>0.786</u>	11.43

(c) Motion diversity, single speaker

Method	APD	Coverage@5 ↑
GT	2.5332	1.0000
CaMN	1.7668	.1284
EMAGE	2.1276	<u>.7261</u>
SemTalk	**2.3857**	.4761
GestureLSM	2.2907	.5506
Ours	<u>2.3759</u>	**.9432**

4.1 Comparison with Prior Methods

DirectBeat achieves the lowest FGD in both BEAT2 protocols (Table 1). Compared with GestureLSM, FGD decreases from 4.088 to 3.631 in the single-speaker setting and from 4.46 to 2.76 in the all-speaker setting, corresponding to reductions of 11% and 38%. BC is competitive in both settings. L1 diversity measures motion amplitude. We complement it with average pairwise distance (APD) and Coverage@5 in the FGD feature space (Table 1c). APD measures the mean Euclidean distance between feature pairs; Coverage@5 measures the fraction of real-feature neighborhoods containing a generated sample (refer to Appendix H for details). DirectBeat achieves APD of 2.3759, close to SemTalk's 2.3857, and the highest generated-motion coverage, 94.32%. These results indicate substantial in-domain motion variation despite the lower L1 score. Section 4.4 further evaluates perceived diversity by a user study and demonstrates that our system produces diverse motion comparable to baselines.

Qualitative comparison. Figure 4 shows comparison of gestures at matched moments in three speech segments. The examples show differences in arm motions, gesture extent, and body posture for the same utterance. For "share the culture," DirectBeat opens both arms outward; for "angered and upset," it raises both hands beside the head. These distinct, speech-appropriate gestures make the illustrated performances more expressive than the smaller arm movements shown by the baselines. The supplementary videos show the complete motion sequences.

4.2 Director and Directive Interface Analysis

Director ablations. Table 2 tests the effectiveness of directives, memory, and corpus knowledge with the Executor fixed. Predicted directives reduce FGD from 3.920 to 3.631; removing memory or corpus knowledge raises it to 3.739 or 3.829, respectively. These results support using recent plans and corpus priors to guide directive generation. Target-motion-derived oracle directives achieve 3.215, indicating scope for further gains from improved planning.

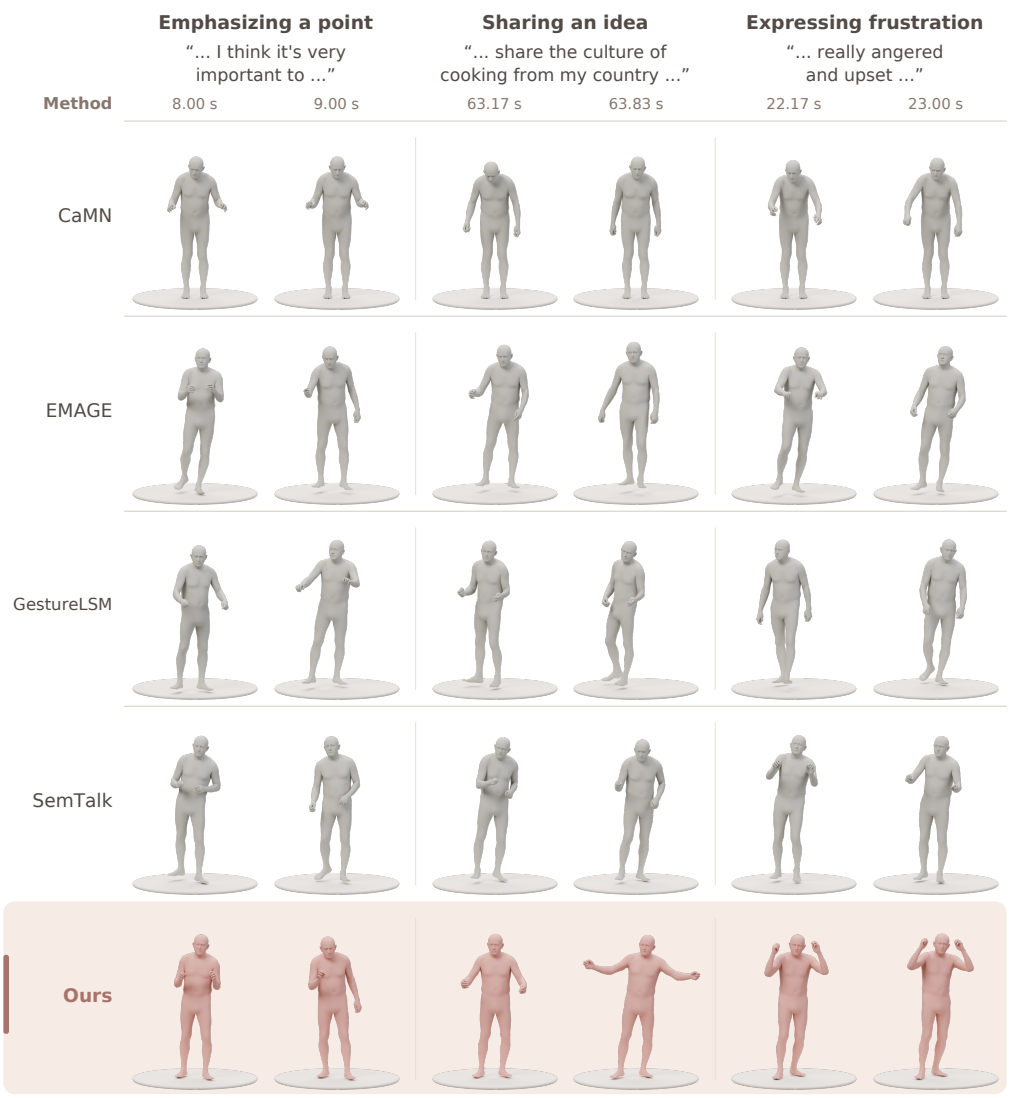

Figure 4: **Qualitative comparison on BEAT2.** Three utterances illustrate emphasis, sharing, and frustration. Each pair of columns shows the same two timestamps across five methods. Supplementary videos show the complete sequences.

Director sensitivity. Table 3 compares Director backends with the Executor fixed. Replacing Gemini with Qwen2.5-Omni-7B or Qwen3-Omni-30B-A3B gives FGD of 3.747 and 3.698, respectively; both improve on generation without directives. The gains across backends support a shared directive interface for different language planners and the same learned Executor. Planning dominates runtime at 4.1–11.4 seconds per two-second window, versus approximately 0.03 seconds for execution. Appendix K gives deployment details and retrieval sensitivity: reducing three transition and four semantic examples to one of each raises FGD from 3.631 to 3.835.

Directive interface: motion retrieval. To evaluate the directive representation, we compare audio, transcripts, and motion-derived directives using the same contrastive objective and motion-encoder architecture. Directives achieve stronger paired-motion retrieval (Figure 5), supporting their use as a motion-specific interface for conditioning the Executor. This comparison complements the generation ablations, which evaluate directives predicted from speech. Appendix D provides the protocol and additional retrieval results.

Table 2: **Director and Executor ablations on single-speaker BEAT2.** Gray denotes target-derived oracle directives.

Configuration	FGD ↓	BC ↑	L1 div.
Ours (Full)	**3.631**	0.781	10.51
w/o Directives	3.920	0.778	10.72
w/o Transcript	3.979	0.781	10.67
w/o Memory	3.739	**0.792**	10.41
w/o Knowledge	3.829	0.775	10.61
Oracle directives	3.215	0.778	10.96

Table 3: **Director replacement with the executor fixed.** Time is seconds per two-second window; deployment details are in Appendix K.

Director	FGD ↓	BC ↑	L1 div.	Time (s)
No directives	3.920	.778	10.72	0
Gemini 3.8 Flash	3.631	.781	10.51	8.6
Qwen2.5-Omni-7B	3.747	.782	10.29	4.1
Qwen3-Omni-30B	3.698	.780	10.09	11.4
Oracle directives	3.215	.778	10.96	–

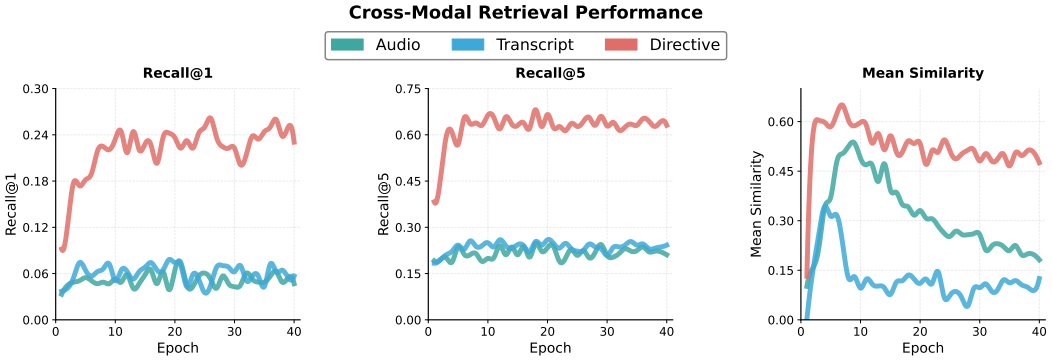

Figure 5: **Cross-modal retrieval.** Motion-derived directives outperform audio and transcripts under a shared contrastive setup. Appendix D gives the protocol.

Table 4: **Warp-VQVAE ablation: reconstruction and generation on single-speaker BEAT2.**

Method	Reconstruction			Generation		
	FGD ↓	BC ↑	L1 div.	FGD ↓	BC ↑	L1 div.
Body-part RVQ	2.612	0.758	12.19	4.912	0.777	11.35
Full-body RVQ	1.935	0.758	11.53	3.752	0.774	10.56
Warp-VQVAE (Ours)	**1.932**	0.758	11.53	**3.631**	**0.781**	10.51

4.3 Executor and Warp-VQVAE Ablations

Executor: transcript synchronization. To test transcript conditioning, we compare absent, unsynchronized, and time-synchronized transcripts. FGD decreases from 3.979 to 3.714 and 3.631, respectively, supporting transcript conditioning aligned with motion-token time. Table 9 in the appendix gives full results.

Warp-VQVAE: motion tokenization. Table 4 compares motion tokenizers in reconstruction and generation. Relative to full-body RVQ, Warp-VQVAE maintains reconstruction fidelity while improving generation FGD from 3.752 to 3.631 and BC from 0.774 to 0.781 with the same Executor. The gains support adpative audio-conditioned latent space temporal warping for generation. The supplementary *rhythm-editing demo* further illustrates this temporal control.

4.4 User Study

We conduct a user study on BEAT2 following the prior protocol on Mturk (Crowston, 2012; Zhang et al., 2025). 33 participants rank synchronized, side-by-side ten-second videos from DirectBeat, EMAGE, SemTalk, and GestureLSM across 89 clips, evaluating realism, semantic consistency, syn-

Table 5: **TED Gesture top-1 preference rates (%) on unseen speech.**

Method	Realism	Semantics	Synchrony	Diversity
EMAGE	31.4	27.9	26.9	32.0
GestureLSM	32.0	33.1	35.6	32.8
Ours	36.6	39.0	37.5	35.2

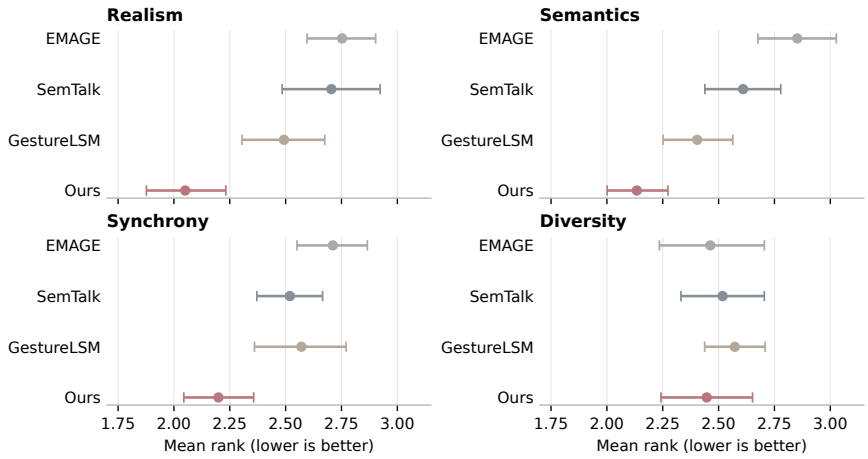

Figure 6: **BEAT2 perceptual comparison.** Mean rank with 95% crossed participant/clip bootstrap intervals; lower is preferred. Each response ranks all four methods.

chrony, and diversity (Figure 6). DirectBeat has the lowest mean rank across all criteria, with larger preference gaps in realism and semantics than in diversity.

To evaluate generalization to unseen speech, we compare DirectBeat with EMAGE and GestureLSM on 100 ten-second TED Gesture clips, with all methods trained on full BEAT2 without speaker-identity conditioning or TED fine-tuning. Twenty-four participants provide at least three rankings per clip. DirectBeat receives the highest top-1 preference across all four criteria, with the largest margin in semantic consistency (Table 5). These results support generalization of the Director–Executor system to unseen speech. The directive interface may facilitate this generalization: the Director expresses new speech contexts as plans in a shared action schema, while the Executor reuses explicit directive–motion correspondences learned from BEAT2.

Supplementary videos. The accompanying webpage (`index.html`) presents 18 videos. A 65-second film follows persona prompt edits through predicted directives to three performances of the same abstract narration (Figure 1). Two public-speech examples, four directive demonstrations, a rhythm-editing example, and ten method comparisons further illustrate the system. The page also includes annotation and director pseudocode.

5 Discussion and Conclusion

DirectBeat uses explicit behavioral directives to connect language-model planning with learned full-body motion generation. BEAT2-Directive provides supervision for this interface, and Warp-VQVAE aligns motion token generation with speech rhythm. The system improves co-speech generation quality on BEAT2 and receives the highest aggregate preference among the compared methods on the unseen TED speech benchmark.

The current system targets offline speech animation. Reducing planning latency is promising for future directions. Using the feedback from the executor to tune the director or the whole system end-to-end for better co-speech generation performance is another promising direction. We plan to release BEAT2-Directive upon acceptance.

AI USE STATEMENT

We used AI tools in the following aspects of this work:

- **System component.** DirectBeat uses a large language model as its Director to generate structured motion directives from speech and context.
- **Dataset annotation.** We used a multimodal large language model to generate initial annotations for BEAT2-Directive. These annotations underwent human filtering, with rejected candidates re-annotated, as described in appendix B.
- **Research and presentation support.** We used coding agents to assist with preliminary experiments to assess early ideas and to create tables, figures, and supplementary videos.
- **Writing support.** We used language models to polish the manuscript and provide suggestions on its presentation.

We have reviewed and validated all AI-assisted work and take responsibility for the final manuscript and accompanying artifacts, including all text and claims.

ETHICS STATEMENT

BEAT2-Directive adds annotations to existing motion-capture recordings. Its distribution and use should respect the licenses of the underlying data and body models. The preference studies involved viewing and ranking generated animations; reported compensation for the BEAT2 study was approximately USD 12 per hour. Generated gestures may express behaviors that the original speaker did not perform or intend, and should not be presented as authentic recordings of that person's behavior. These limitations matter when applying the system to embodied agents or representations of real people.

REPRODUCIBILITY STATEMENT

Section 3 describes the system and its conditioning inputs. Appendices B, K, and G give the annotation, planning, training, and decoding procedures. Metric definitions, ablations, and perceptual protocols appear in Appendices H, I.1, J, and L. The supplementary webpage includes annotation and director pseudocode, together with 18 videos of the complete pipeline, behavioral control, and method comparisons. We plan to release BEAT2-Directive upon acceptance.

REFERENCES

Simon Alexanderson, Gustav Eje Henter, Taras Kucherenko, and Jonas Beskow. Style-controllable speech-driven gesture synthesis using normalising flows. *Computer graphics forum*, 39(2):487–496, 2020.

Tenglong Ao, Qingzhe Gao, Yuke Lou, Baoquan Chen, and Libin Liu. Rhythmic gesticulator: Rhythm-aware co-speech gesture synthesis with hierarchical neural embeddings. *ACM Transactions on Graphics (TOG)*, 41(6):1–19, 2022.

Tenglong Ao, Zeyi Zhang, and Libin Liu. Gesturediffuclip: Gesture diffusion model with clip latents. *ACM Transactions on Graphics (TOG)*, 42(4):1–18, 2023.

Paul Boersma. Praat, a system for doing phonetics by computer. *Glot. Int.*, 5(9):341–345, 2001.

Judee K Burgoon, Thomas Birk, and Michael Pfau. Nonverbal Behaviors, Persuasion, and Credibility. *Human communication research*, 17(1):140–169, 1990.

Justine Cassell, Hannes Högni Vilhjálmsson, and Timothy Bickmore. Beat: the behavior expression animation toolkit. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 477–486, 2001.

Bohong Chen, Yumeng Li, Yao-Xiang Ding, Tianjia Shao, and Kun Zhou. Enabling synergistic full-body control in prompt-based co-speech motion generation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 10, New York, NY, USA, 2024a. ACM. doi: 10.1145/3664647.3680847.

Junming Chen, Yunfei Liu, Jianan Wang, Ailing Zeng, Yu Li, and Qifeng Chen. DiffSHEG: A Diffusion-Based Approach for Real-Time Speech-driven Holistic 3D Expression and Gesture Generation, 2024b. URL https://arxiv.org/abs/2401.04747.

Qingrong Cheng, Xu Li, and Xinghui Fu. Siggesture: Generalized co-speech gesture synthesis via semantic injection with large-scale pre-training diffusion models. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11, 2024.

chuan guo, Yuxuan Mu, Xinxin Zuo, Peng Dai, Youliang Yan, Juwei Lu, and Li Cheng. Generative human motion stylization in latent space. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=daEqXJ0yZo.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, 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.

Kevin Crowston. Amazon mechanical turk: A research tool for organizations and information systems scholars. In *Shaping the Future of ICT Research. Methods and Approaches: IFIP WG 8.2, Working Conference, Tampa, FL, USA, December 13-14, 2012. Proceedings*, pp. 210–221. Springer, 2012.

Jan P De Ruiter, Adrian Bangerter, and Paula Dings. The Interplay Between Gesture and Speech in the Production of Referring Expressions: Investigating the Tradeoff Hypothesis. *Topics in cognitive science*, 4(2):232–248, 2012.

Fengyi Fang, Sicheng Yang, and Wenming Yang. CoordSpeaker: Exploiting gesture captioning for coordinated caption-empowered co-speech gesture generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30761–30771, 2026.

Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. Momask: Generative masked modeling of 3d human motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1900–1910, 2024.

Ikhsanul Habibie, Weipeng Xu, Dushyant Mehta, Lingjie Liu, Hans-Peter Seidel, Gerard Pons-Moll, Mohamed Elgharib, and Christian Theobalt. Learning speech-driven 3d conversational gestures from video. In *Proceedings of the 21st ACM international conference on intelligent virtual agents*, pp. 101–108, 2021.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 1501–1510, 2017.

Yiming Huang, Weilin Wan, Yue Yang, Chris Callison-Burch, Mark Yatskar, and Lingjie Liu. Como: Controllable motion generation through language guided pose code editing. In *European Conference on Computer Vision*, pp. 180–196. Springer, 2024.

Daniel Kahneman. *Thinking, fast and slow*. macmillan, 2011.

Jing Li, Di Kang, Wenjie Pei, Xuefei Zhe, Ying Zhang, Zhenyu He, and Linchao Bao. Audio2gestures: Generating diverse gestures from speech audio with conditional variational autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11293–11302, 2021a.

Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. AI Choreographer: Music Conditioned 3D Dance Generation with AIST++. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13401–13412, 2021b.

Haiyang Liu, Naoya Iwamoto, Zihao Zhu, Zhengqing Li, You Zhou, Elif Bozkurt, and Bo Zheng. DisCo: Disentangled Implicit Content and Rhythm Learning for Diverse Co-Speech Gestures Synthesis. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 3764–3773, 2022a.

Haiyang Liu, Zihao Zhu, Naoya Iwamoto, Yichen Peng, Zhengqing Li, You Zhou, Elif Bozkurt, and Bo Zheng. BEAT: A Large-Scale Semantic and Emotional Multi-Modal Dataset for Conversational Gestures Synthesis. *arXiv preprint arXiv:2203.05297*, 2022b.

Haiyang Liu, Zihao Zhu, Giorgio Becherini, Yichen Peng, Mingyang Su, You Zhou, Xuefei Zhe, Naoya Iwamoto, Bo Zheng, and Michael J Black. Emage: Towards unified holistic co-speech gesture generation via expressive masked audio gesture modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1144–1154, 2024.

Pinxin Liu, Haiyang Liu, Luchuan Song, and Chenliang Xu. Intentional gesture: Deliver your intentions with gestures for speech, 2025a. URL https://arxiv.org/abs/2505.15197.

Pinxin Liu, Luchuan Song, Junhua Huang, Haiyang Liu, and Chenliang Xu. Gesturelsm: Latent shortcut based co-speech gesture generation with spatial-temporal modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10929–10939, 2025b.

David McNeill. Hand and Mind. *Advances in Visual Semiotics*, 351, 1992.

Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *International conference on machine learning*, pp. 7176–7185. PMLR, 2020.

Kunkun Pang, Dafei Qin, Yingruo Fan, Julian Habekost, Takaaki Shiratori, Junichi Yamagishi, and Taku Komura. Bodyformer: Semantics-guided 3d body gesture synthesis with transformer. *ACM Transactions on Graphics (TOG)*, 42(4):1–12, 2023.

Mathis Petrovich, Michael J. Black, and Gül Varol. Tmr: Text-to-motion retrieval using contrastive 3d human motion synthesis, 2023. URL https://arxiv.org/abs/2305.00976.

Ekkasit Pinyoanuntapong, Pu Wang, Minwoo Lee, and Chen Chen. MMM: Generative Masked Motion Model. In *CVPR*, pp. 1546–1555, 2024.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Sicheng Yang, Zunnan Xu, Haiwei Xue, Yongkang Cheng, Shaoli Huang, Mingming Gong, and Zhiyong Wu. Freetalker: Controllable speech and text-driven gesture generation based on diffusion models for enhanced speaker naturalness. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7945–7949. IEEE, 2024.

Hongwei Yi, Hualin Liang, Yifei Liu, Qiong Cao, Yandong Wen, Timo Bolkart, Dacheng Tao, and Michael J Black. Generating Holistic 3D Human Motion from Speech. In *CVPR*, 2023.

Youngwoo Yoon, Bok Cha, Joo-Haeng Lee, Minsu Jang, Jaeyeon Lee, Jaehong Kim, and Geehyuk Lee. Speech Gesture Generation from the Trimodal Context of Text, Audio, and Speaker Identity. *ACMTOG*, 39(6), 2020.

Xiangyue Zhang, Jianfang Li, Jiaxu Zhang, Ziqiang Dang, Jianqiang Ren, Liefeng Bo, and Zhigang Tu. Semtalk: Holistic co-speech motion generation with frame-level semantic emphasis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13761–13771, 2025.

Zeyi Zhang, Tenglong Ao, Yuyao Zhang, Qingzhe Gao, Chuan Lin, Baoquan Chen, and Libin Liu. Semantic gesticulator: Semantics-aware co-speech gesture synthesis, 2024. URL https://arxiv.org/abs/2405.09814.

APPENDIX CONTENTS

A Additional Related Work **15**

B BEAT2-Directive Annotation Pipeline **15**

C Supplementary Videos and Qualitative Examples **16**

D Cross-Modal Retrieval Experiments **17**

D.1 Motivation 17

D.2 Contrastive Learning Framework 17

D.3 Evaluation Metrics 18

D.4 Results and Discussion 18

E Directive-Conditioned Motion Generator **18**

E.1 Condition Injection 19

E.2 Temporally Synchronized Positional Embeddings 19

E.3 Motion Masked Transformer 19

E.4 Training Objective 19

F Warp-VQVAE Details **19**

G Implementation Details **20**

H Metric Details **20**

I Additional Motion-Generation Ablations **22**

I.1 Warp-VQVAE: Motion Tokenization 22

I.2 Executor: Transcript Synchronization 22

J User Study Details **23**

J.1 Task Setup and Instructions 23

J.2 Evaluation Criteria 24

J.3 Quality Control and Filtering 24

K Director Specification **24**

K.1 Director Sensitivity and Runtime 25

L Evaluation on Unseen TED Speech **25**

M Additional Evaluation Details **26**

M.1 Evaluation Resources 26

M.2 Planning and Execution Controls 26

M.3 Perceptual Aggregation 26

A Additional Related Work

Behavior selection and motion execution. Co-speech generation has developed from rule-based animation (Cassell et al., 2001) to learned speech-conditioned synthesis (Ao et al., 2022; Liu et al., 2024; Yi et al., 2023; Chen et al., 2024b). Semantic gesture retrieval and semantic-aware generation (Zhang et al., 2024; 2025; Cheng et al., 2024) already introduce behavioral information into this process. InG (Liu et al., 2025a) also connects speech-to-intention reasoning with a motion generator. These are relevant precedents for a planning-and-execution view. Our system predicts dense, body-specific plans and realizes them through an executor trained on aligned motion descriptions. A shared directive schema links speech-conditioned planning to supervised physical execution.

Language-conditioned motion. GestureDiffuCLIP (Ao et al., 2023) and FreeTalker (Yang et al., 2024) condition co-speech motion on additional text or prompts. SynTalker (Chen et al., 2024a) and CoordSpeaker (Fang et al., 2026) combine speech with supplied motion descriptions (often agnostic to speech) for full-body control. SynTalker learns implicit prompt labels through motion–text alignment, whereas CoordSpeaker generates multi-granular training captions offline from motion and uses a hierarchically conditioned latent diffusion model. Text-to-motion work such as CoMo (Huang et al., 2024) likewise uses explicit intermediate descriptions for controllability. Our directives similarly provide an explicit intermediate representation. Here, the director predicts the sequential directives from speech and recent behavioral context, and the executor realizes them with transcript conditioning and Warp-VQVAE's audio-conditioned temporal warping. Our retrieval study characterizes motion correspondence, while generation ablations evaluate the resulting planning–execution interface.

Motion content and temporal control. Style-conditioned synthesis can use feature modulation such as AdaIN (chuan guo et al., 2024; Huang & Belongie, 2017; Ao et al., 2023). DisCo (Liu et al., 2022a) investigates decomposing speech-related content and rhythm, while Rhythmic Gesticulator (Ao et al., 2022) explicitly normalizes and denormalizes temporal actions around acoustic events. Warp-VQVAE builds on this temporal-control perspective by learning resampling fields in a quantized motion representation. The paired resampling fields connect motion tokenization with acoustic alignment during decoding.

B BEAT2-Directive Annotation Pipeline

We render BEAT2 motion and divide it into two-second windows. The annotation model is `gemini-3.8-flash`, with rendered frames sampled at 3 FPS (six per window) and decoding temperature 0.1. Speech and transcripts are excluded from the annotation input so descriptions are based on visible motion. The annotations are then aligned with the corresponding speech, yielding 101,660 triplets.

Five experts in co-speech motion generation check whether each candidate matches the visible motion, obeys the schema, separates left- and right-arm actions, and avoids unsupported descriptions. Rejected candidates are re-annotated. The initial screening pass rate is 92.5% under this filtering procedure. Descriptions focus on visible physical actions, and the schema includes an unclassified category.

Actionability and body-relative anchoring. The annotation prompt requests physical actions rather than abstract summaries of the speaker's emotion or communicative purpose. Relative anatomical landmarks, such as "hand toward the left shoulder" or "elbow near the ribs," reduce reliance on estimating absolute 3D coordinates from rendered views. These descriptions specify action structure while allowing variation in the realized trajectory.

Bilateral descriptions and categorical fields. The JSON schema separates right-arm action, left-arm action, and global posture to represent asymmetric gestures. Enumerated motion phases distinguish preparation, active motion, hold, and retraction. The gesture classes are listed in Table 6. The unclassified category accommodates movements whose communicative function is ambiguous. Table 7 illustrates how the schema captures bilateral actions and posture across motion phases.

Table 6: BEAT2-Directive annotation schema. Motion-state proportions are reported within the annotation set. Gesture classes are listed without proportions.

Dimension	Categories or fields
Gesture class	Beat, deictic, iconic, metaphoric, emblem, none/unclassified.
Motion phase	Active (79.85%), preparation (0.62%), retraction (8.07%), hold (11.46%).
Bilateral action	Separate `right_arm_action` and `left_arm_action` descriptions.
Body posture	Torso orientation, lean, stance, and weight distribution when visible.
Alignment	Two-second interval and associated transcript, attached after annotation.

Table 7: Additional annotation examples. Descriptions are condensed from annotated motion clips.

Clip / time (s)	State / class	Observed body and bilateral action
`4_lawrence` 10–12	Active / metaphoric	Right arm sweeps upward and outward toward shoulder height; left arm stays lower; slight forward/left torso lean.
`9_miranda` 10–12	Active / deictic	Right hand rises to a pointing shape near the chest; left hand briefly lifts near the waist then lowers.
`12_zhao` 12–14	Retraction / metaphoric	Both bent arms extend downward toward the thighs while the body returns toward an upright stance.
`30_katya` 10–12	Active / metaphoric	Right hand sweeps inward and upward toward the shoulder; left arm remains outward at waist height with palm up.

C Supplementary Videos and Qualitative Examples

The supplementary videos extend the main-paper comparison in Figure 4. Open `index.html` to view the interactive webpage. Table 8 summarizes its 18 videos across five visual sections, followed by the annotation and Director pseudocode. Chapter shortcuts navigate between stages of the persona demonstration and between the two audio passes of the rhythm example, and a tabbed viewer provides access to all ten method comparisons.

Inspecting and editing behavior. The 65-second film uses saved prompts, predicted directives, and generated motions to show the complete planning and execution pipeline. Prompt edits request a preference for right-hand gestures or wider, open-palm gestures. The corresponding directive excerpts show how these preferences change bilateral action descriptions. The final scene presents three personas performing the same abstract narration, making the behavioral effect visible while speech remains fixed. The four directive clips provide shorter examples of the action-to-motion correspondence.

Editing speech rhythm. The rhythm example illustrates the division of labor between directives and Warp-VQVAE (Section 3.4): directives and motion tokens determine what the body does, while decoding-time warping determines when it happens. We synthesize a second narration of the closing Jensen Huang passage from the identical transcript with a different phrase tempo and pause structure. The saved DirectBeat motion tokens are then re-decoded into a re-rhythmed motion. The two performances are shown side by side, with the audio waveform and the current word displayed above each avatar.

Unseen speech. Two examples animate public speech recordings with character avatars. These demonstrate motion generation from speech; the avatars are not reconstructions of the speakers' captured body motion. The TED evaluation in Appendix L complements these case studies with a perceptual comparison on unseen speech.

Table 8: **Supplementary video contents.** The webpage accompanies the paper and includes synchronized speech playback.

Section	Videos	Demonstrated capability
Persona walkthrough	1 (65 s)	Prompt edit → predicted directive → motion; three performances of the same abstract narration.
Public speakers	2	Speech-driven animation using recordings of Jensen Huang and Fei-Fei Li.
Directive execution	4	Right-arm sweep, outward two-hand motion, emphatic downward gesture, and expanded gesture space.
Rhythm editing	1 (21 s)	Same transcript with a different delivery rhythm; gesture content is kept while its timing follows the new speech (illustrative re-timing).
Method comparisons	10	CaMN, EMAGE, GestureLSM, SemTalk, and ours under matched rendering.

D Cross-Modal Retrieval Experiments

D.1 Motivation

We compare the correspondence of audio, transcripts and motion-derived directives with their paired motion. This diagnostic measures the motion-specific information provided by each conditioning modality. Generation ablations complement it by evaluating directives predicted from speech.

D.2 Contrastive Learning Framework

We train a separate symmetric contrastive model for each pairing of motion with audio rhythm features, transcripts, or annotated directives. The audio representation contains amplitude-envelope, short-time energy, and onset features resampled to 30 FPS. Transcript text comes from the word-aligned speech annotations; directive text concatenates the body-posture and bilateral arm-action descriptions.

The motion input comprises normalized 6D rotations for the 55 SMPL-X joints, giving 330 features per frame; root translations are excluded from retrieval. Motion and audio each pass through a bidirectional GRU with 256 hidden units per direction and attention pooling. Both text modalities use a frozen pretrained DistilBERT encoder with attention-mask-weighted mean pooling. Each modality is then projected to 256 dimensions by a two-layer MLP with a 512-dimensional hidden layer, GELU activation, and dropout of 0.1. The resulting embeddings are L2-normalized for cosine-similarity retrieval (Petrovich et al., 2023).

The network is optimized using a symmetric contrastive loss to pull aligned motion–condition pairs (audio, transcript, or directive) together while pushing misaligned pairs apart:

$$\mathcal{L}_{\text{ret}} = \frac{1}{2}(\mathcal{L}_{m\to c} + \mathcal{L}_{c\to m}) \tag{4}$$

where the motion-to-condition loss is defined as:

$$\mathcal{L}_{m\to c} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp\left(\alpha\, \mathbf{m}_i \mathbf{c}_i^\top\right)}{\sum_{j=1}^{N} \exp\left(\alpha\, \mathbf{m}_i \mathbf{c}_j^\top\right)} \tag{5}$$

Here, $\mathbf{m}_i$ and $\mathbf{c}_i$ denote the L2-normalized embeddings for the motion and the corresponding condition modality, respectively. N is the training batch size, and $\alpha = \min(\exp(s), 100)$ is a learned similarity scale, with s initialized to 2.6593. The condition-to-motion loss $\mathcal{L}_{c\to m}$ is computed symmetrically.

For retrieval, windows from the BEAT2 training partition are divided in dataset order: the first 90% are used for contrastive training and the remaining 10% for held-out evaluation. Each model is trained for 80 epochs with AdamW, an initial learning rate of 1×10^{-3}, weight decay of 1×10^{-4}, cosine learning-rate decay, and a batch size of 32. Gradients are clipped to a norm of 1.0. Training batches are shuffled; evaluation batches retain dataset order, and incomplete batches are dropped in both cases.

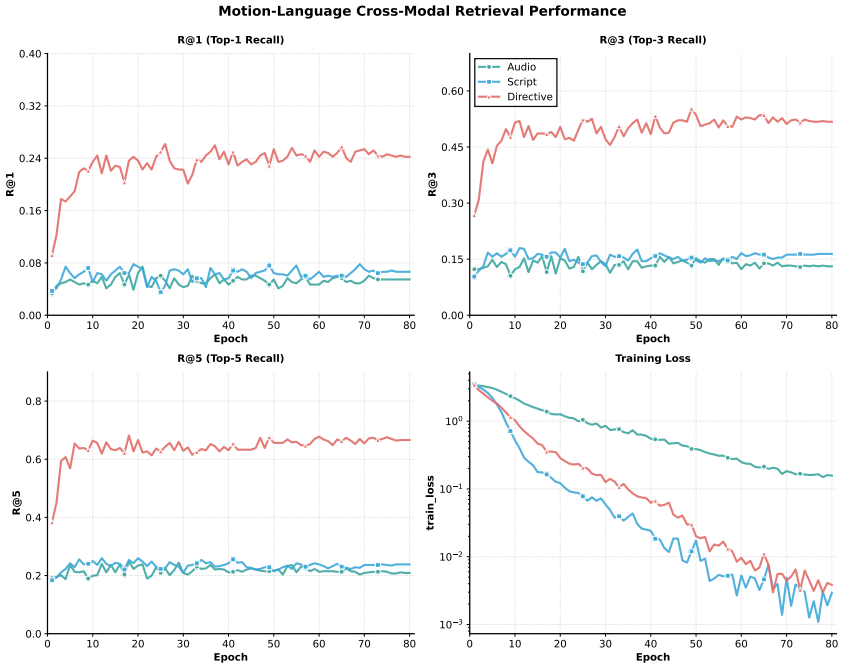

Figure 7: **Additional cross-modal retrieval results.** Test recall and training loss over 80 epochs.

D.3 Evaluation Metrics

On the held-out split, recall at K (R@K) measures the proportion of queries whose paired motion or condition appears among the top K candidates:

$$\text{R@K} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \mathbb{1}(\text{rank}_i \leq K), \tag{6}$$

where N_{test} is the number of evaluated queries. Retrieval ranks candidates within each evaluation batch of 32, and recall is averaged over motion-to-condition and condition-to-motion queries. The random R@5 reference is $5/32 = 15.625\%$. We report R@1, R@3, and R@5; the evaluator also records the mean cosine similarity of paired embeddings.

D.4 Results and Discussion

Figure 7 compares held-out retrieval for audio, transcripts, and motion-derived directives under the same contrastive objective and motion-encoder architecture. Directives achieve higher recall across the reported ranks, supporting their use as motion-specific supervision for the Executor. The generation ablations in Section 4.2 complement this correspondence test by evaluating directives predicted from speech.

E Directive-Conditioned Motion Generator

The executor predicts discrete motion tokens $\mathbf{z}$ conditioned on structured directives D and timed transcripts W. Here, D includes the motion descriptions, motion-state labels c_s, and gesture-class labels c_g. Warp-VQVAE subsequently uses audio to align the decoded motion with speech. We use conditional masked-token prediction following Motion Masked Transformer (Guo et al., 2024; Pinyoanuntapong et al., 2024).

E.1 Condition Injection

The text in D and transcript W are encoded by separate lightweight text encoders and injected through alternating cross-attention layers: directive attention supplies behavioral content, while transcript attention supplies temporally grounded speech evidence. During training, conditions are independently dropped and replaced by null embeddings, reducing over-reliance on any single input and enabling classifier-free guidance (Ho & Salimans, 2022) at inference.

E.2 Temporally Synchronized Positional Embeddings

TextGrid provides word-level speech intervals, whereas BERT-style tokenization may split one word into multiple subword pieces. We preserve timing by assigning each transcript token an integer position on the latent motion-token grid. In the implementation, these aligned indices are passed as `transcript_embd_positions`; the transcript stream then selects positional encodings from the same sinusoidal basis used by the motion-token stream.

For example, suppose TextGrid marks the spoken word `NeurIPS` over a single pronunciation interval, and the center of this interval maps to the discrete latent motion frame k. If BERT tokenizes this word into several pieces, such as `neu`, `##ri`, and `##ps`, all three subword tokens are assigned the same position index k. They therefore receive the same temporal code as the motion token at frame k, rather than being treated as three consecutive time steps. In this way, transcript attention is grounded in the same temporal coordinate system as motion-token prediction, without introducing raw audio features into the executor.

E.3 Motion Masked Transformer

We first tokenize the human motion sequence $M \in \mathbb{R}^{T_m \times 333}$ using our pretrained Warp-VQVAE with residual vector quantization. Here, T_m counts motion frames; each frame contains 330 rotation features and three root-translation coordinates. For the base layer, write the discrete token sequence as $\mathbf{z} = (z_1, \ldots, z_T)$, where T is the latent sequence length. During training, we adopt a bidirectional masked transformer architecture. We randomly mask a subset of the motion tokens and train the network to reconstruct them. The masking ratio follows $\cos(\pi u/2)$ for a randomly sampled level $u \in [0, 1]$ (Guo et al., 2024).

E.4 Training Objective

The generator p_θ predicts a categorical distribution over the codebook. Training minimizes the negative log-likelihood of masked tokens given the visible tokens, directives, and transcript:

$$\mathcal{L}_{\text{mask}} = \sum_{i \in \mathcal{S}} -\log p_\theta(z_i \mid \widetilde{\mathbf{z}}, D, W) \tag{7}$$

where $\widetilde{\mathbf{z}}$ is the masked token sequence and $\mathcal{S}$ is the set of masked positions. Conditioning on D includes its text and categorical labels c_s, c_g.

F Warp-VQVAE Details

Warp-VQVAE uses temporal warping to adjust motion dynamics before quantization and reintroduce audio-conditioned rhythm afterward.

Rhythm Normalization Given a motion sequence M, the encoder produces $Z_e = E(M)$. The normalization block applies Equation (2) to these features and audio A, giving $Z_n = \text{Warp}_{\text{norm}}(Z_e, A)$. Its learned sampling grid normalizes timing before quantization, prioritizing structural motion content.

Latent Temporal Bottleneck The normalized latent Z_n is quantized to token indices $\mathbf{z}$ with codebook embeddings $Z_q = \mathcal{C}(\mathbf{z})$. During training, random temporal resampling perturbs Z_q before rhythm alignment, using the interpolation operator in Equation (2) with a randomly perturbed grid. This perturbation discourages the codebook from storing residual timing information and encourages it to represent motion structure.

Rhythm Denormalization and Decoding The rhythm block applies the same operation to codebook embeddings Z_q and audio A, using its own predictor to align motion timing with speech. The decoder reconstructs motion as $\hat{M} = F(\mathrm{Warp}_{\mathrm{rhythm}}(Z_q, A))$. At generation time, Decode_ψ first looks up the embeddings $\mathcal{C}(\mathbf{z})$ of the predicted tokens, then applies this rhythm block and decoder.

G Implementation Details

Warp Operator The interpolation operator Interp in Equation (2) is implemented with PyTorch's `torch.nn.functional.grid_sample`, using differentiable bilinear interpolation over the temporal latent grid. In the supplied implementation, both warps use $\Delta_{\max} = 3$, `padding_mode=border`, and `align_corners=True`. The grid offset is bounded by $3/T$, corresponding to $3(T-1)/(2T)$ latent indices. The offset bound does not enforce a monotonic sampling grid.

Training Configuration We train the motion models on a single NVIDIA RTX 4090 GPU using AdamW with a learning rate of 0.0002. Warp-VQVAE is trained with a batch size of 256 for 400 epochs. The quantization layer is implemented as a residual quantizer comprising 6 codebooks with 512 codes each, a latent dimension of 512, and a dropout rate of 0.5. The directive-to-motion generator is trained for 100,000 iterations with a batch size of 128. The conditioning dropout rate is set to 0.1 for each condition independently.

Inference Pipeline During inference, we first query the director with the source audio and transcript to obtain the full sequence of directives. The director processes the audio and transcript in two-second windows, retaining at most two preceding directive windows for local continuity. The directive-to-motion generator then uses the directives and transcripts to generate motion tokens in successive two-second windows. Executor runtime is approximately 0.03 seconds per two-second motion window. Planning is performed separately; Appendix K gives the end-to-end runtime of the offline pipeline.

H Metric Details

Fréchet Gesture Distance Fréchet Gesture Distance (FGD) (Yoon et al., 2020) compares generated and reference motion distributions in the feature space of a pretrained motion encoder. It is the squared Wasserstein-2 distance between Gaussian approximations to these distributions; lower values indicate closer feature distributions:

$$\mathrm{FGD}(R, G) = \|\mu_r - \mu_g\|^2 + \mathrm{Tr}\left(\Sigma_r + \Sigma_g - 2\left(\Sigma_r \Sigma_g\right)^{1/2}\right), \tag{8}$$

where R and G are the real and generated feature sets, with respective means μ_r, μ_g and covariance matrices Σ_r, Σ_g.

L1 Diversity L1 diversity accumulates absolute deviations from each clip's temporal mean, divided by the total number of frames. For flattened joint coordinates $p_{c,t}$ and clip mean $\bar{p}_c$, its definition is

$$\mathrm{L1div} = \frac{\sum_c \sum_{t=1}^{T_c} \|p_{c,t} - \bar{p}_c\|_1}{\sum_c T_c}. \tag{9}$$

This statistic measures within-clip motion amplitude. Feature-space APD complements it by measuring pairwise distances across motion embeddings. Both are computed under the corresponding benchmark preprocessing and joint representation.

Feature-space evaluation protocol APD and Coverage@5 use the same pretrained motion encoder as FGD. We evaluate the 15 shared single-speaker test recordings, truncating each recording to the common prefix available for all methods and the reference. At 30 FPS, the 55 SMPL-X joint rotations are converted to a 330-dimensional 6D rotation representation. The FGD evaluator trims each sequence to a multiple of 32 frames and extracts 240-dimensional temporal features using the frozen `AESKConv_240_100` checkpoint. We pool these features across recordings, yielding

1,760 samples per method and for the reference. Distances are computed directly in this feature space without additional normalization. Let $R = \{r_i\}_{i=1}^{N_r}$ and $G = \{g_j\}_{j=1}^{N_g}$ denote the real and generated feature samples.

Average Pairwise Distance (APD) APD measures feature dispersion by averaging Euclidean distances over all distinct, unordered pairs of generated samples:

$$\mathrm{APD}(G) = \frac{2}{N_g(N_g-1)} \sum_{1 \le i < j \le N_g} \|g_i - g_j\|_2. \tag{10}$$

We compute the exact all-pairs mean rather than estimating it from randomly sampled pairs. The GT row uses the same calculation on R. Larger APD indicates broader feature dispersion; the GT value provides a reference for the scale of natural motion variation.

Coverage@5 We use the coverage metric of Naeem et al. (2020) with $k = 5$. For each real feature r_i, let ρ_i be its Euclidean distance to its fifth-nearest *other* real feature, excluding the self-match. Coverage is the fraction of these real-feature neighborhoods containing at least one generated sample:

$$\mathrm{Coverage@5}(R, G) = \frac{1}{N_r} \sum_{i=1}^{N_r} \mathbb{1}\left[\min_{1 \le j \le N_g} \|r_i - g_j\|_2 \le \rho_i\right]. \tag{11}$$

Our implementation includes samples on the neighborhood boundary. Coverage lies in $[0, 1]$ and is reported as a percentage in the tables, with higher values indicating broader coverage of the reference feature distribution. The GT row has coverage one because the reference set is evaluated against itself. Together, APD and Coverage@5 characterize the spread of generated motion features and their coverage of real motion.

Beat Constancy Beat Constancy (BC) (Li et al., 2021b) measures alignment between detected speech onsets and motion beats, defined as local minima of upper-body joint velocities excluding the fingers. A Gaussian kernel scores the temporal distance from each motion beat to its nearest audio beat:

$$\mathrm{BC} = \frac{1}{|\mathcal{B}_m|} \sum_{t_m \in \mathcal{B}_m} \exp\left(-\frac{\min_{t_a \in \mathcal{B}_a} (t_m - t_a)^2}{2\sigma^2}\right), \tag{12}$$

where $\mathcal{B}_m$ and $\mathcal{B}_a$ are the sets of motion-beat and audio-beat times, respectively, and σ controls the Gaussian alignment window in the same time units.

Table 9: **Executor ablation: transcript synchronization on single-speaker BEAT2.**

Configuration	FGD ↓	BC ↑	L1 div.
w/o Transcript	3.979	**0.781**	10.67
Transcript w/o sync	3.714	0.778	10.58
Transcript w/ sync (Ours)	**3.631**	**0.781**	10.51

I Additional Motion-Generation Ablations

I.1 Warp-VQVAE: Motion Tokenization

Table 4 in the main paper compares body-part RVQ, full-body RVQ, and Warp-VQVAE in reconstruction and downstream generation. Full-body RVQ improves both reconstruction and generation FGD over body-part RVQ. Warp-VQVAE preserves reconstruction fidelity while further improving generation FGD and BC. These results support full-body tokenization and audio-conditioned re-timing as complementary parts of motion execution.

I.2 Executor: Transcript Synchronization

Table 9 tests transcript conditioning and the synchronized positional embeddings in Section 3.3. The no-transcript condition retains directives and audio conditioning in Warp-VQVAE; the other conditions add transcript tokens with either sequential or time-aligned positions. Adding unsynchronized transcripts reduces FGD from 3.979 to 3.714, and mapping subword tokens onto the motion-token timeline further reduces it to 3.631. The synchronized configuration also recovers BC from 0.778 to 0.781, matching the no-transcript condition. The FGD gains support both transcript content and explicit temporal alignment, while BC remains similar across the three settings.

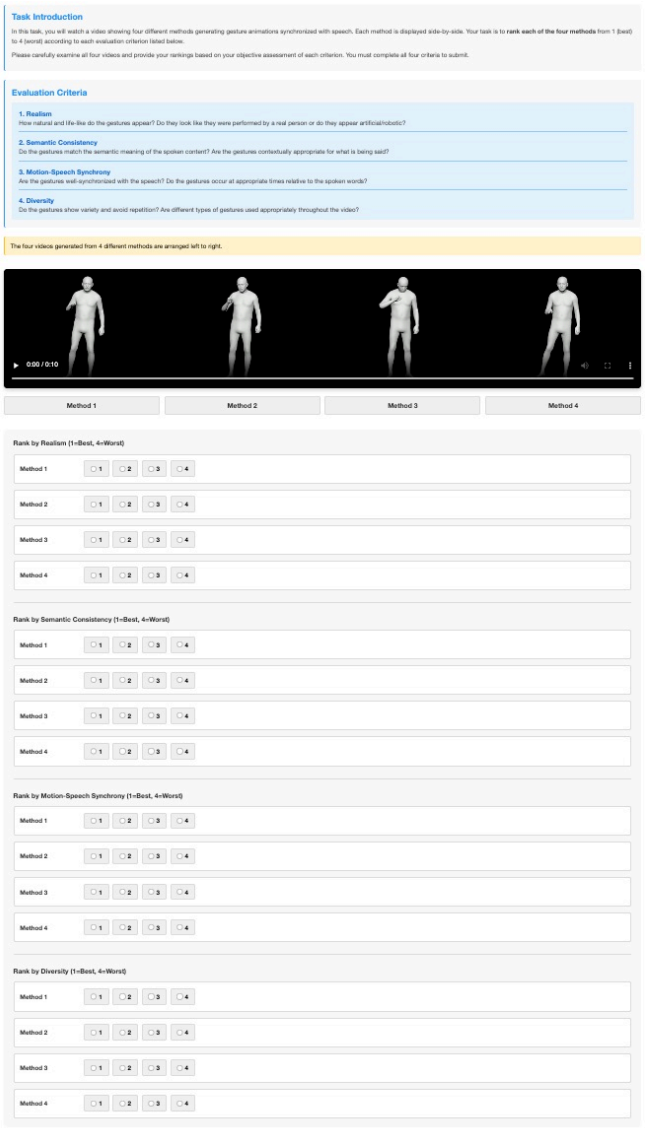

Figure 8: **User Study Portal Interface.** Participants are presented with four side-by-side generated videos and asked to rank them across four perceptual criteria.

J User Study Details

The BEAT2 perceptual study was conducted on Amazon Mechanical Turk (MTurk) (Crowston, 2012). This section describes the interface, instructions, and filtering procedure.

J.1 Task Setup and Instructions

Participants viewed four synchronized, side-by-side 10-second videos from DirectBeat, EMAGE (Liu et al., 2024), SemTalk (Zhang et al., 2025), and GestureLSM (Liu et al., 2025b) (Figure 8). The portal provided the following instructions:

> *In this task, you will watch a video showing four different methods generating gesture animations synchronized with speech. Each method is displayed side-by-side. Your task is to* ***rank each of the four methods*** *from 1 (best) to 4 (worst) according to each evaluation criterion listed below. Please carefully examine all*

four videos and provide your rankings based on your objective assessment of each criterion. You must complete all four criteria to submit.

J.2 Evaluation Criteria

Participants ranked all four methods without ties for each criterion. The interface defined the criteria as follows (Zhang et al., 2025):

1. **Realism:** How natural and life-like do the gestures appear? Do they look like they were performed by a real person or do they appear artificial/robotic?
2. **Semantic Consistency:** Do the gestures match the semantic meaning of the spoken content? Are the gestures contextually appropriate for what is being said?
3. **Motion-Speech Synchrony:** Are the gestures well-synchronized with the speech? Do the gestures occur at appropriate times relative to the spoken words?
4. **Diversity:** Do the gestures show variety and avoid repetition? Are different types of gestures used appropriately throughout the video?

J.3 Quality Control and Filtering

Submissions were screened using the recorded completion time for each Human Intelligence Task (HIT). Responses completed too quickly to view the 10-second clips and assess all four criteria were excluded. Appendix M.3 describes the aggregation of complete rankings.

K Director Specification

The director processes a sequence of speech windows $X_t = (A_t, W_t, \tau_t)$, a knowledge summary $\mathcal{K}$, retrieved examples $\mathcal{E}_t \subset \mathcal{E}$ from the exemplar bank, and a fixed schema. The director operates at temperature 0.15 and maintains a history H_t of at most two prior directives. Here, τ_t denotes the time interval of window t. The schema includes time, phase, class, posture, bilateral actions, and a short rationale. The director additionally permits a neutral state and an unclassified gesture label; the annotation schema uses the four motion phases listed above. The auxiliary rationale explains the selected behavioral plan.

Corpus preparation. Annotated BEAT2-Directive sequences form an example bank containing current transcript/action fields and preceding state/class fields. An IDF index supports transcript retrieval. A corpus summary provides state/class priors, state transitions, arm-use patterns, class-associated cue words, and continuity rules. Appendix M.1 describes the available resource metadata.

Window-level procedure.

1. Tokenize current transcript W_t.
2. Rank examples by IDF-weighted lexical overlap and compatibility with the preceding state, class and arm-use pattern.
3. Retrieve three transition-oriented and four semantic examples, removing duplicates.
4. Construct a prompt containing A_t, W_t, τ_t, the last two directives, corpus summary, transition hints and retrieved examples.
5. Generate a schema-constrained directive. Preserve the input interval and transcript in the output record.
6. Add the physical directive fields to memory and retain the most recent two windows.

The memory record contains time, transcript, motion state, gesture class, posture and left/right actions. Only the two most recent records are retained. The supplementary pseudocode specifies prompt construction, schema fields, scoring and history updates. Backend names identify the evaluated model versions.

Table 10: Retrieval sensitivity to the number of transition and semantic exemplars.

Transition examples	Semantic examples	FGD ↓	BC ↑	L1 diversity
3	4	3.631	.781	10.51
1	1	3.835	.780	10.48
0	0	3.829	.775	10.61

Table 11: Runtime estimates, in seconds per two seconds of generated motion. API and local deployments are identified explicitly; rendering time is excluded.

Method / director	Planning	End-to-end
EMAGE	0	.14
GestureLSM	0	.05
Executor without directives	0	.03
gemini-3.8-flash	8.6	8.63
Qwen2.5-Omni-7B, local	4.1	4.13
Qwen3-Omni-30B-A3B-Instruct, API	11.4	11.43

K.1 Director Sensitivity and Runtime

Retrieval sensitivity. Table 10 varies the numbers of transition and semantic examples while keeping the other Director and Executor components fixed. The standard three-plus-four configuration achieves FGD of 3.631, compared with 3.835 for one example of each type and 3.829 without examples. BC and L1 diversity vary little across these settings. The benefit of the larger example set supports using a range of behavior and transition examples; adding only two examples performs similarly to omitting retrieval.

Director replacement. Table 3 in the main paper changes only the Director backend, holding the prompt, memory, retrieval, schema, Executor, decoder, and evaluation protocol fixed. Relative to the no-directive FGD of 3.920, Qwen2.5 and Qwen3 retain 59.9% and 76.8% of the Gemini improvement. Both alternatives improve generation with the same Executor, supporting the directive interface across language-model backends. For deployment, `gemini-3.8-flash` uses an API, Qwen2.5-Omni-7B runs locally, and Qwen3-Omni-30B-A3B-Instruct is an open-source model accessed through an API.

Runtime. Table 11 compares planning and end-to-end runtime for two-second motion windows. Execution takes approximately 0.03 seconds, while the Director adds 4.1–11.4 seconds. Local Qwen2.5 reduces end-to-end runtime from 8.63 to 4.13 seconds relative to the Gemini configuration. Planning therefore dominates latency and is the main target for improving the speed of the offline system. These timings include deployment differences between local inference and APIs.

L Evaluation on Unseen TED Speech

TED Gesture protocol. The out-of-distribution study uses 100 randomly sampled 10-second clips with speakers and speech content unseen during training. Each audio input is converted to 16 kHz WAV and transcribed using Whisper `base.en`; word timestamps produce an aligned TextGrid. EMAGE and GestureLSM were retrained on full BEAT2 with speaker-identity conditioning removed from all three methods, without TED fine-tuning. Motions use the same standard SMPL-X shape and rendering settings. Twenty-four participants rank anonymized videos in randomized order, with at least three judgments per clip.

Results. Table 5 in the main paper reports top-1 preference rates under this matched protocol. DirectBeat receives the highest preference across all four criteria, with the largest margin in semantic consistency. The results support generalization of the Director–Executor system to unseen speech without TED fine-tuning.

Table 12: **Planning and execution controls.**

Condition	Executor	FGD ↓	BC ↑
Shuffled predicted plans	Ours	5.891	0.772
Retrieval-only plans	Ours	3.80	0.890
Predicted plans	Ours	3.631	0.781
Target-derived plans (oracle)	Ours	3.215	0.778
Same predicted plans	SynTalker	5.792	0.716

M Additional Evaluation Details

M.1 Evaluation Resources

The single-speaker metadata contains 118 recordings and 9,842 windows: 95/8/15 recordings and 7,569/834/1,439 windows for training, validation, and testing, respectively. Recording IDs are disjoint across these splits. The 15 speech-query recordings match the test-recording set. The exemplar bank contains 95 sequences and 7,569 windows, indexed by sequence and window. Corpus priors and IDF statistics are derived from the same bank.

M.2 Planning and Execution Controls

Plan and executor comparisons. Table 12 varies the plans with our Executor fixed, then compares our Executor with SynTalker (Chen et al., 2024a) using the same predicted plans. Predicted directives improve FGD over shuffled and retrieval-only plans, and target-derived oracle directives reduce it further. With the same predicted plans, our Executor also achieves lower FGD and higher BC than SynTalker. These comparisons support both speech-conditioned planning and a dedicated directive-to-motion Executor. Retrieval-only plans achieve the highest BC but poorer FGD than predicted plans, showing that beat alignment alone does not capture motion-distribution fidelity.

M.3 Perceptual Aggregation

Figure 6 summarizes the complete four-method rankings from the BEAT2 study described in Appendix J. For each criterion, $r_{i,m}$ is the rank (1–4) assigned to method m in response i. With common response weights w_i, the mean ranks $\bar{r}_m$ satisfy

$$\sum_m \bar{r}_m = \frac{\sum_i w_i \sum_m r_{i,m}}{\sum_i w_i} = 10. \tag{13}$$

The ranking summary contains 873 complete judgments across 33 participant IDs and 89 clip IDs; 72 clips have ten judgments and 17 have nine. Each criterion uses the same complete rankings and denominator.

DirectBeat has the lowest mean rank across all four criteria, with larger preference gaps in realism and semantic consistency than in diversity. The rankings support perceptual improvements alongside the distributional metrics in Table 1.

Intervals use 5,000 crossed bootstrap replicates. Participant IDs and clip IDs are independently resampled; each complete response receives the product of their multiplicities, keeping all four method rankings paired. The resulting 95% intervals account for repeated participants and clips. Full-precision means preserve the rank-sum identity.