MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Plan and Double-Check: Streaming Multimodal Q-Former for Online Robot Action Generation

Hori, Chiori; Korekata, Ryosuke; Kambara, Motonari; Masuyama, Yoshiki; Jain, Siddarth; Corcodel, Radu; Romeres, Diego; Le Roux, Jonathan

TR2026-136 September 29, 2026

Abstract

Integrating humanoid robots into daily life demands effective collaboration, where robots must accurately perceive human actions and context to work toward common objectives. Previous studies generate robot action sequences for human instructional videos, but most rely on offline processing with pre-segmented clips. In real-world interaction, however, robots must handle unsegmented multimodal inputs in a streaming manner. In this paper, we extend a Q-Former-based robot action generation framework to streaming processing. Our method extracts audiovisual features from short temporal chunks, incorporates leftcontext information into query embeddings, and incrementally generates action sequences and confirmation messages using a large language model. By carefully designing attention masks, the model can be trained efficiently in parallel, similar to offline methods. Experiments show that our streaming approach achieves low latency with less than 10% accuracy degradation compared to offline processing.

Interspeech 2026

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Plan and Double-Check: Streaming Multimodal Q-Former for Online Robot Action Generation

Chiori Hori [1,**], *Ryosuke Korekata* [1,2], *Motonari Kambara* [1,2], *Yoshiki Masuyama* [1], *Siddarth Jain* [1], *Radu Corcodel* [1], *Diego Romeres* [1], *Jonathan Le Roux* [1]

[1] Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA
[2] Keio University, Yokohama, Kanagawa, Japan
{chori, masuyama, leroux}@merl.com

Abstract

Integrating humanoid robots into daily life demands effective collaboration, where robots must accurately perceive human actions and context to work toward common objectives. Previous studies generate robot action sequences for human instructional videos, but most rely on offline processing with pre-segmented clips. In real-world interaction, however, robots must handle unsegmented multimodal inputs in a streaming manner. In this paper, we extend a Q-Former-based robot action generation framework to streaming processing. Our method extracts audio-visual features from short temporal chunks, incorporates left-context information into query embeddings, and incrementally generates action sequences and confirmation messages using a large language model. By carefully designing attention masks, the model can be trained efficiently in parallel, similar to offline methods. Experiments show that our streaming approach achieves low latency with less than 10% accuracy degradation compared to offline processing.

Index Terms: human-robot interaction, online robot action planning, robot confirmation generation, streaming multimodal Q-former, multimodal LLM

1. Introduction

Humanoid robots capable of dexterous manipulation are widely regarded as promising platforms for supporting everyday human activities. To achieve this, effective human–robot collaboration requires robots to perceive and interpret human actions and environmental context to achieve shared goals. Research on robot action planning has recently been spurred by leveraging multimodal scene understanding trained from human demonstration videos leveraging multimodal transformer architectures that unify perception, language, and action.

Demonstration-driven vision–language models learn to generate robot actions directly from visual and linguistic inputs [1–4], with some approaches incorporating additional sensor embeddings [5]. These methods map multimodal observations and task instructions to executable action sequences, effectively embedding planning within an end-to-end policy. In contrast, **LLM-based symbolic and open-world planning approaches** [6–9] treat language models as high-level planners that generate compositional, text-based task plans through language reasoning. These methods target open-world and partially observable settings and often employ iterative re-prompting to refine plan executability [7]. While they demonstrate strong compositional and semantic generalization, the resulting plans are typically abstract and require additional grounding modules to execute on physical robots. More recent **Vision–Language–Action (VLA) models** incorporate stronger reasoning and temporal modeling [10–12]. Audio features are also added [13–15]. In contrast to symbolic LLM planners that require separate grounding mechanisms, VLA models embed perception and action within a single embodied framework. Within this family, hierarchical VLA approaches further emphasize open-ended instruction following by autonomously decomposing language into executable action sequences that are directly carried out by the robot [16].

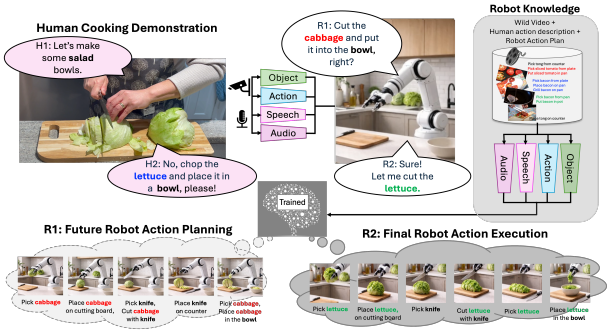

Figure 1: *Multimodal large language models generate robot action confirmations together with the micro-step action sequences, enabling transparent communication and verification of the robot's* ***planned future actions*** *before they are executed.*

Separately, **Multimodal Scene Understanding** methods that integrate diverse modalities such as audio, visual signals, and text using multimodal language models have advanced scene understanding for acquiring task knowledge from human demonstration videos. Early work on multimodal scene-understanding-based robot action planning leveraged audio-visual transformers trained on cooking videos [17, 18]. In these approaches, multimodal representations of human action steps were derived from demonstration videos and mapped to executable robot micro-actions, enabling task-level planning grounded in multimodal perception. Furthermore, these works adopt a **Q-former-style fusion** mechanism, originally developed for efficient vision language alignment [19] and demonstrated to be more efficient than conventional fusion methods [20, 21], and embed it within a multimodal action planning architecture to extract task-relevant representations for explicit step-level generation and confirmation.

Despite extensive research on robot action planning, reliable real-world execution remains challenging due to distribution shifts and unforeseen conditions. To reduce errors, **Human-in-the-loop replanning** methods have been proposed [22, 23], allowing robots to correct plans through dialogue. However, these approaches are reactive and intervene only af-

** indicates the corresponding author.

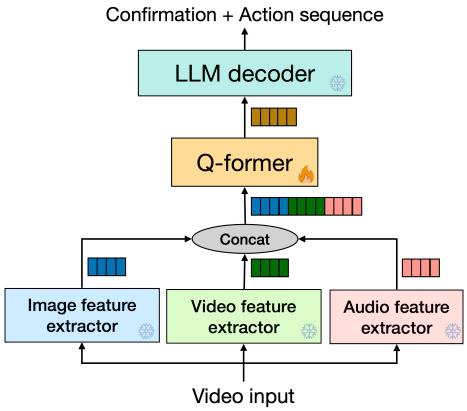

Figure 2: *Q-former-based confirmation sentence generation and action planning model [24]. AVBLIP-based action generation with a Q-former. The generated embeddings are fed to the LLM decoder.*

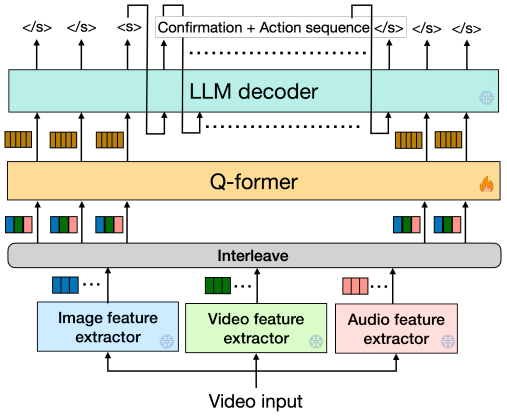

Figure 3: *Online streaming Q-former-based confirmation sentence generation and action planning model. AVBLIP-based action generation is performed in a streaming manner, where chunk-wise multi-modal features are interleaved and converted to query embeddings using the Q-former. The embeddings are used to detect begin token "⟨s⟩" and the following token sequence, or end token "⟨/s⟩" that indicates no more tokens from the current chunk.*

ter failures occur. One approach integrates confirmation generation directly into multimodal planning, enabling proactive human validation before execution and reducing failure risk at the planning stage as shown in Fig. 1 [24, 25]. Importantly, it maintains multimodal grounding as in demonstration-driven methods while introducing explicit step-level planning and confirmation generation.

Building upon this line of research, this paper proposes **Streaming Multimodal Q-Former**, a real-time framework that integrates streaming multimodal grounding with confirmation-aware action planning in a low-latency manner.

2. Action Planning Data

The YouCook2 dataset [26] contains textual annotations that explain human cooking activities in a step-by-step manner [26]. The original human cooking instruction is adopted as the robot action description. The human instructions were mapped into micro-step action sequences, allowing a single-arm robot to achieve task objectives equivalent to those demonstrated by humans [24].

a) Robot Action Description: For example, in a video demonstrating the preparation of porcini mushroom risotto, an instruction is annotated as "Add the chopped porcini mushrooms to the pan and stir them with the rice until evenly mixed," with timestamps from 01:10 to 01:35.

b) Robot Micro-step Action: The micro-step action sequence for a single-arm robot is structured using four elements: "single-arm action," "target object," "preposition," and "place." A predefined set of 12 primitive single-arm actions is employed: Open, Close, Pick, Place, Pour, Stir, TurnOn, TurnOff, Wipe, Cut, Scoop, Squeeze. The target objects are selected, whenever possible, from the nouns appearing in the corresponding human action instructions. As an illustration, the instruction described above can be decomposed into the following sequence of micro-steps: *Pick (spoon), Scoop (porcini mushrooms) from (cutting board), Place (porcini mushrooms) into (pan), Stir (rice and mushrooms) in (pan).*

3. Robot action generation using AVBLIP

This work employs AVBLIP [18], where BLIP-2 [19], a vision-language pre-training method, is extended to handle multimodal features. BLIP-2 bootstraps from a frozen image encoder and a frozen LLM, where a Querying Transformer (Q-former) [27] is trained to bridge the gap between the vision and text modalities. The image encoder in BLIP-2 is replaced with audio-visual encoders for video, audio, and text feature sequences in AVBLIP. Figure 2 illustrates the AVBLIP architecture consisting of a Q-former and an LLM decoder. The Q-former is trained to extract a fixed number of embeddings from a multimodal encoder that outputs sequences of different lengths. The self-attention layers are shared between two transformer submodules: (1) a multimodal transformer that interacts with the frozen audio-visual encoders and (2) a text transformer that works as a text encoder and a text decoder. A set of learnable query embeddings is input to the multimodal transformer. The queries interact with each other through self-attention layers and with audio-visual features through cross-attention layers. The queries can additionally interact with text through the same cross-attention layers. Finally, the queries are converted to an output feature.

The LLM Decoder generates a sequence of micro-step actions for a single-arm robot from multimodal features aligned to language features obtained by the Q-former. The LLM Decoder is constructed with a frozen LLM and a feed-forward layer. Using the LLM as a decoder leverages the LLM's inference capabilities when generating action sequences. In this study, we use OPT-2.7B [28] as the LLM.

The training of AVBLIP consists of two stages: (1) multimodal-language representation learning with frozen multimodal encoders and (2) multimodal-to-language generative learning with a frozen LLM. In the second stage, we connect the Q-former to the frozen LLM Decoder and perform multimodal action sequence generation. As shown in Fig. 2, the extracted multimodal features from the video clip are converted to token embeddings using the Q-former. The embedding vectors are then projected to the LLM embedding space using a fully-connected layer. Then, the LLM Decoder generates action sequences and descriptions (confirmation sentences) for the given video clip. We use the cross-entropy loss function for the ground-truth sequences in this stage.

4. Online streaming AVBLIP

4.1. Architecture

In this paper, we extend the AVBLIP-based system architecture to process the video input in a streaming manner and output a response sentence and action sequence at any appropriate time. The original AVBLIP model in Fig. 2 generates a response sequence for a short video clip of around 10-20 seconds by extracting image, video, and audio features, concatenating them, and feeding the result to the Q-former and the LLM decoder. In this case, we need to provide the exact beginning and end time of the video clip. On the other hand, the proposed streaming AVBLIP system in Fig. 3 generates the output sequence by extracting the multimodal features from each short chunk (e.g., 1 second) and interleaving them to feed the result to the Q-former and the LLM. This can be repeated over sequential chunks in a streaming manner, and the LLM predicts the begin-of-sequence and the following tokens, or the end-of-sequence token that indicates no more tokens are generated from the current chunk. In this way, the LLM decides the right timing to output the response based on the chunk-wise query embeddings generated by the Q-former, where the model does not require explicit video clip segmentation.

Interleaving and chunking approaches have already been applied to LLM-based streaming automatic speech recognition (ASR) [29, 30]. These approaches alternately perform audio encoding and token generation frame by frame or chunk by chunk to achieve low-latency ASR. We essentially follow those ideas, but extend them for the AVBLIP framework so they can work with the Q-former without retraining the LLM decoder.

4.2. Alignment

To train the streaming model, we need to provide alignment information between the input chunks and the target tokens. One reasonable alignment is that all target tokens are aligned to the last chunk of the video clip, according to the manual annotation of the training data. Furthermore, we can promote the model to output the response earlier than the end of clip to achieve low-latency responses. However, unlike speech recognition or speech translation, it is difficult to estimate precise frame-to-token alignments using CTC and Transducer models. For conversation tasks, real human-to-human conversation data with utterance-level annotation are used to train the model to respond to the user in natural response timing. However, there is no sufficient data with plausible annotations for human-to-robot dialog and action generation.
In this work, we investigate two approaches:

Sampling: During training, we sample a response time for each video clip by randomly selecting a chunk earlier than the last chunk in the clip, where we avoid selecting aggressively early chunks by enforcing an allowable maximum shift.

Loss-based selection: We sample a response time using the sampling method and compute the loss function. We further compute the loss for the last chunk of the clip. If the sampled loss is smaller or sufficiently close to the last loss, we backprop from the sampled loss; otherwise, we do so from the last loss. This method requires more computation than the simple sampling, but we could avoid irrelevant response timing by considering the loss difference. We apply a fixed threshold $\tau(> 0)$ to adopt the sampled loss based on $(\mathcal{L}_{\text{smpl}} - \mathcal{L}_{\text{orig}}) < \tau$, where $\mathcal{L}_{\text{smpl}}$ and $\mathcal{L}_{\text{orig}}$ denote the sampled and last losses after length normalization.

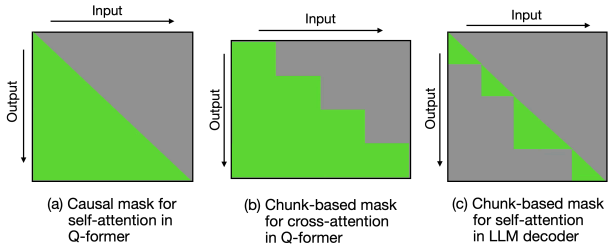

Figure 4: *Attention masks for training the online Q-former. The green area allows attention between input (keys) and output (queries) while the gray area blocks the attention.*

Table 1: *YouCook2 data [26].*

	#Videos	#Clips	Duration/clip [sec]
Training	1173	8743	19.7
Validation	416	3117	19.8

4.3. Attention mask

In training, we can efficiently parallelize forward and backward propagation for chunk-wise sequence generation by designing attention masks appropriately. Figure 4 shows examples of attention masks used for training the online streaming AVBLIP model. In each mask, the X-axis shows input sequence corresponding to the key/value and Y-axis shows output sequence corresponding to the query.

In Fig. 4, (a) Causal mask is used for self-attention in the Q-former and the original LLM decoder, (b) Chunk-based mask is used for cross-attention in the Q-former, where each query token in a chunk can attend to the past chunks to make the query embeddings to be left-context dependent, not attending to the future chunks, and (c) Chunk-based mask for self-attention in the streaming LLM decoder, where the LLM attends to only the query embeddings and the generated tokens in the current chunk. This example assumes the response is generated from the third chunk.

4.4. Decoding

Sequence generation is performed by chunk-based decoding using the streaming Q-former and the LLM decoder. The Q-former receives multi-modal features chunk by chunk and outputs query embeddings by attending to the current and all cached previous hidden states. The LLM decoder generates a token sequence auto-regressively from the query embeddings of the current chunk, and proceeds to the next chunk if it predicts the end-of-sequence token. In this streaming process, we use greedy decoding to output the predicted tokens immediately.

5. Experiments

5.1. Setup

We evaluated the proposed method using the YouCook2 dataset [26], which consists of cooking action video clips aligned with human action instruction in natural language. The statistics of the dataset are summarized in Table 1. We used the validation set for evaluation since the test set annotation was not publicly available, splitting the validation set in half and applying cross-validation to obtain the final results.

We utilized micro-step action sequences [24], which contain 2,790 unique action phrases of 195 verbs, 2,229 objects, 33 prepositions, and 1002 places. Multimodal features such

Table 2: *Comparison of action sequence and response quality on different model configurations.*

Model	Alignment in training	Generation	Average latency [sec]	Miss detection rate [%]	Action sequence		Action description	
					BLEU-2	METEOR	BLUE-2	METEOR
(1) Baseline	N/A	offline	-	-	0.357	0.251	0.221	0.154
(2) w/ interleave	N/A	offline	-	-	0.354	0.248	0.219	0.152
(3) Streaming	clip end	offline	-	-	0.356	0.246	0.216	0.154
(4) Streaming	clip end	online	–0.78 ($\pm$2.41)	5.20	0.321	0.223	0.197	0.138
(5) Streaming	sampling	online	–2.12 ($\pm$2.72)	4.80	0.319	0.215	0.194	0.138
(6) Streaming	loss-based	online	**–2.92** ($\pm$2.20)	**4.40**	**0.328**	**0.228**	**0.212**	**0.145**
[25]	N/A	offline	-	-	0.370	0.260	0.229	0.159

as video, image, and audio were extracted using Omnivore [31], Contrastive Language-Image Pre-Training (CLIP) [32], and Audio Spectrogram Transformer (AST) [33], respectively. The features were then projected and concatenated to a single multimodal feature sequence before feeding them to the Q-former. In the streaming scenario, we apply this process for each one-second chunk. The dimensions of the audio, image, and video features were 527, 768, and 3806, respectively.

In prior studies [24,25], subtitles in the video were also used to improve the performance. However, we did not use them since the subtitles are typically not available in online streaming scenarios. Although it is possible to employ a streaming ASR system to feed the partial transcript, this makes the system complex and the evaluation process complicated. We will consider this extension in future work.

The Q-former was initialized with the pre-trained weights of BERTbase [34], while the cross-attention layers were randomly initialized. The number of dimensions of the hidden layers was set to 768 and the number of queries was set to 16, which resulted in 188M parameters in total. We finetuned the Q-former with the frozen LLM decoder for both offline and online scenarios, respectively. For alignment sampling, the allowable maximum shift was set to 5 seconds prior to the clip's end position. For loss-based selection, the threshold τ was set to 0.002 because our preliminary experiments showed relatively high performance around this value.

The performance was evaluated using the BLEU-2 and METEOR scores computed between the generated and ground-truth sequences as used in the robotics field [35,36]. We applied cross-validation, where one half of the validation set was used to select the best-epoch model, and the other half was used to measure model performance. Thus, the shown scores are the average over the two subsets.

Furthermore, we simulated streaming use case by appending $5-10$ seconds of preceding and succeeding video frames to each video clip and evaluated whether the model can correctly generate action sequences and description in a reasonable timing. The appended margin length was randomly selected and sometimes shortened if there is an overlap with the previous or next video clip. To evaluate the streaming performance, we measured average latency ($\pm$ standard deviation) and miss detection rate. The average latency is the time at which the model starts to generate an action sequence, relative to the end time of the original clip. The miss detection rate is the proportion of clips for which the model did not generate any action sequence.

5.2. Results

Table 2 shows the quality of action sequences and descriptions generated by the baseline and streaming Q-former-LLM systems. "(1) Baseline" denotes the Q-former model trained with pairs of concatenated multimodal features and their target sequences without streaming extension. Compared to the prior work in [25], BLEU and METEOR scores were slightly worse. This is mainly because we used greedy decoding while they used beam search decoding with beam size 5. "(2) w/ interleave" shows the result when we interleaved the multimodal features into one-second chunks but trained the model in the offline setting similar to the baseline. Compared to "(1) Baseline," the differences are negligible, confirming that interleaving the features does not have negative impact on the performance. "(3) Streaming" shows the result of the streaming Q-former used as an offline model, where the model was forced to generate the sequence from the chunk at the original clip ending time. We do not see significant differences in the performance metrics, meaning that the streaming model was trained successfully without losing the performance for offline use.

The results of streaming Q-formers from "(4)" to "(6)" correspond to the models trained to start generating the target sequence at clip end, sampled chunk, and selected chunk by CE loss, respectively. Compared to the baseline, we observed a certain degradation in BLUE and METEOR scores. This is mainly because the model incorrectly started to generate responses attending to irrelevant portion of the video clip, and in some cases, the model did not output any tokens other than the end-of-sequence token. However, the degradations are less than 10% relative in all sequence quality metrics. Generally, accuracy degradation due to streaming processing is unavoidable, but a 10% drop is still within an acceptable range. Furthermore, these models show lower average latency compared with the baseline, ranging from 1 to 3 seconds, which is important for low-latency streaming systems. Among the three streaming models, the model with loss-based alignment demonstrated the best performance across all metrics. Also, the standard deviation of latency shows that the loss-based method selects output time with less fluctuation. This indicates that the alignment should be selected to avoid increasing the loss during training.

6. Conclusions

This paper proposed a Q-Former-based robot action generation framework that enables streaming processing. Our method extracts audio-visual features from short temporal chunks, incorporates left-context information into query embeddings, and incrementally generates action sequences and confirmation messages using a large language model. By carefully designing attention masks, the model can be trained efficiently in parallel, similar to offline methods. Experiments show that our streaming approach achieves low latency with less than 10% accuracy degradation compared to offline processing. We also investigated three different methods to provide alignment information to the model during training, and demonstrated that the loss-based alignment selection works better than other approaches.

7. References

[1] A. Brohan, N. Brown, J. Carbajal *et al.*, "RT-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[2] ——, "RT-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[3] M. Ahn, A. Brohan, N. Brown *et al.*, "Do as I can, not as I say: Grounding language in robotic affordances," in *Proc. CoRL*, 2023.

[4] M. Shridhar, L. Manuelli, and D. Fox, "CLIPort: What and where pathways for robotic manipulation," in *Proc. CoRL*, 2022.

[5] D. Driess, F. Xia, M. S. M. Sajjadi *et al.*, "PaLM-E: An embodied multimodal language model," in *Proc. ICML*, 2023.

[6] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "ProgPrompt: Generating situated robot task plans using large language models," in *Proc. ICRA*, 2023.

[7] L. Sun, D. K. Jha, C. Hori, S. Jain, R. Corcodel, X. Zhu, M. Tomizuka, and D. Romeres, "Interactive planning using large language models for partially observable robotic tasks," in *Proc. ICRA*, 2024.

[8] Y. Ding, X. Zhang, S. Amiri, N. Cao, H. Yang, A. Kaminski, C. Esselink, and S. Zhang, "Integrating action knowledge and LLMs for task planning and situation handling in open worlds," *Autonomous Robots*, vol. 47, no. 8, pp. 981–997, 2023.

[9] S. S. Raman, V. Cohen, E. Rosen, I. Idrees, D. Paulius, and S. Tellex, "Planning with large language models via corrective re-prompting," in *NeurIPS Foundation Models for Decision Making Workshop*, 2022.

[10] X. Zhao *et al.*, "CoT-VLA: Visual chain-of-thought reasoning for vision-language-action models," in *Proc. CVPR*, 2025.

[11] C.-P. Huang, Y.-H. Wu, M.-H. Chen, Y.-C. F. Wang, and F.-E. Yang, "ThinkAct: Vision-language-action reasoning via reinforced visual latent planning," in *Proc. NeurIPS*, 2025.

[12] H. Zhou, C. Ma, and G. H. Lee, "VLA-4D: Embedding 4D awareness into vision-language-action models for spatiotemporally coherent robotic manipulation," *arXiv preprint arXiv:2511.17199*, 2025.

[13] H. Akbari, L. Yuan, R. Qian, W.-N. Chuang, S. Chang, Y. Cui, F. Ashraf, and B. Gong, "VATT: Transformers for multimodal self-supervised learning from raw video, audio and text," in *Proc. NeurIPS*, 2021.

[14] Z. Liu, C. Chi, E. Cousineau, N. Kuppuswamy, B. Burchfiel, and S. Song, "Maniwav: Learning robot manipulation from in-the-wild audio-visual data," *arXiv preprint arXiv:2406.19464*, 2024.

[15] X. Wei, H. Zhang, X. Cao, S. Xie, W. Ge, Y. Li, and C. Wang, "Audio-VLA: Adding contact audio perception to vision-language-action models for robotic manipulation," *arXiv preprint arXiv:2511.09958*, 2025.

[16] L. X. Shi, B. Ichter, M. Equi, L. Ke, K. Pertsch, Q. Vuong, J. Tanner, A. Walling, H. Wang *et al.*, "Hi Robot: Open-ended instruction following with hierarchical vision-language-action models," in *Proc. ICML*, 2025.

[17] C. Hori, P. Peng, D. Harwath, X. Liu, K. Ota, S. Jain, R. Corcodel, D. Jha, D. Romeres, and J. Le Roux, "Style-transfer based speech and audio-visual scene understanding for robot action sequence acquisition from videos," in *Proc. Interspeech*, 2023, pp. 4663–4667.

[18] M. Kambara, K. Sugiura, S. Khurana, K. Ota, S. Jain, R. Corcodel, D. Jha, D. Romeres, J. Le Roux, and C. Hori, "Human action understanding-based robot planning using multimodal LLM," in *Proc. ICRA Workshop for "Cooking Robotics: Perception and motion planning"*, 2024.

[19] J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *Proc. ICML*, 2023.

[20] J. An, J. Lee, J. Lee, and Y. Son, "Towards LLM-centric multimodal fusion: A survey on integration strategies and techniques," *arXiv preprint arXiv:2506.04788*, 2025.

[21] S. N. Wadekar, A. Chaurasia, A. Chadha, and E. Culurciello, "The evolution of multimodal model architectures," *arXiv preprint arXiv:2405.17927*, 2024.

[22] P. Sharma, B. Sundaralingam, V. Blukis, C. Paxton, T. Hermans, A. Torralba, J. Andreas, and D. Fox, "Correcting robot plans with natural language feedback," in *Proc. RSS*, 2022.

[23] E. Merlo, M. Lagomarsino, and A. Ajoudani, "A human-in-the-loop approach to robot action replanning through LLM common-sense reasoning," *IEEE RA-L*, vol. 10, no. 10, pp. 10 767–10 774, 2025.

[24] C. Hori, M. Kambara, K. Sugiura, K. Ota, S. Khurana, S. Jain, R. Corcodel, D. Jha, D. Romeres, and J. Le Roux, "Interactive robot action replanning using multimodal LLM trained from human demonstration videos," in *Proc. ICASSP*, 2025.

[25] C. Hori, Y. Masuyama, S. Jain, R. Corcodel, D. K. Jha, D. Romeres, and J. Le Roux, "Robot confirmation generation and action planning using long-context Q-former integrated with multimodal LLM," in *Proc. ASRU*, 2025.

[26] L. Zhou, C. Xu, and J. J. Corso, "Towards Automatic Learning of Procedures from Web Instructional Videos," in *Proc. AAAI*, 2018.

[27] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. ECCV*, 2020.

[28] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan *et al.*, "OPT: Open pre-trained transformer language models," *arXiv preprint arXiv:2205.01068*, 2022.

[29] F. Seide, M. Doulaty, Y. Shi, Y. Gaur, J. Jia, and C. Wu, "Speech ReaLLM–real-time streaming speech recognition with multimodal llms by teaching the flow of time," *arXiv preprint arXiv:2406.09569*, 2024.

[30] J. Jia, G. Keren, W. Zhou, E. Lakomkin, X. Zhang, C. Wu, F. Seide, J. Mahadeokar, and O. Kalinli, "Efficient streaming LLM for speech recognition," in *Proc. ICASSP*, 2025.

[31] R. Girdhar, M. Singh, N. Ravi, L. van der Maaten, A. Joulin, and I. Misra, "Omnivore: A single model for many visual modalities," in *Proc. CVPR*, 2022.

[32] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. ICML*, 2021, pp. 8748–8763.

[33] Y. Gong, Y.-A. Chung, and J. Glass, "AST: Audio Spectrogram Transformer," in *Proc. Interspeech*, 2021, pp. 571–575.

[34] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. ACL*, 2018, pp. 4171–4186.

[35] A. Nguyen, D. Kanoulas, L. Muratore, D. G. Caldwell, and N. G. Tsagarakis, "Translating videos to commands for robotic manipulation with deep recurrent neural networks," in *Proc. ICRA*, 2018.

[36] X. Xu, K. Qian, B. Zhou, S. Chen, and Y. Li, "Two-stream 2D/3D residual networks for learning robot manipulations from human demonstration videos," in *Proc. ICRA*, 2021.