

# Automated Context-Aware Navigation Support for Individuals with Visual Impairment Using Multimodal Language Models in Urban Environments

Alton Chao\*

Erika Maquiling

Edmund Chao

Roshan Sanjeev

Tonko Bossen

Ross Greer

## Abstract

Vision transformer capabilities for images have increased significantly in recent years. Multimodal vision transformers are now able to generate accurate captions for images and demonstrate strong capabilities in understanding this visual input. More recently, these models have been built to handle videos, with or without audio. However, these transformers have seldom been trained on datasets related to accessibility. In this study, we focus on generating navigation instructions for individuals with visual impairment in the context of outdoor, urban environments. We use the spatial-temporal vision language model (VLM), VideoLLaMA3, to process videos and generate a series of instructions based on a prompt specifically designed for individuals with visual impairments. With our approach, we were able to surpass the performance of the GPT-4o model. In the future, we anticipate this approach being extended through the use of landmark detection and improved fine-tuning. In this work, we investigate the use of VLMs as a backbone within a pipeline that incorporates prompting, postprocessing, and other techniques to develop spatially and temporally accurate instructions.

## 1. Introduction

In the United States, more than 1 in 4 adults have a disability, with 5.5% of adults and 625,000 children in the U.S [1] having blindness or serious difficulty seeing, even with glasses [2]. The rise of transformers and VLMs presents a valuable opportunity to leverage these technologies to create accessibility-driven tools [3]. For example, audio language transformers can improve the interpretative capabilities of smart assistants for individuals with dysarthria [4]. However, the current large-scale datasets used in these models are not extended to people with disabilities, such as wheelchair users or people with guide dogs [5, 6]. In this paper, we focus on the capabilities of VLMs in generating



Figure 1. We aim to provide accessible technology aimed at assisting people with visual impairments in independently navigating dynamic outdoor environments, such as the famous Shibuya crossing above, known for its busyness and pedestrian traffic.

navigation instructions for people with visual impairments. We specifically aim to assist pedestrians with visual impairments in navigating dynamic outdoor environments. This lies in providing safe, accurate, time-efficient, and easy-to-follow instructions [7], as illustrated in Figure 1. We aim to spark a broader dialogue on accessibility in the research community and to forge new pathways that bridge computer vision innovations with assistive technologies.

## 2. Related Work

We provide a method specifically designed to caption videos for people with visual impairments, expanding upon previous research in the accessibility space. In this section, we describe related research on the use of VLMs for image and video captioning and instruction generation. In particular, we discuss some of the existing literature on using machine learning to generate navigation instructions.

### 2.1. Vision Language Models

VLMs bridge the connection between large language models (LLMs) and computer vision [8]. Through the use of image tokens, visual data can be passed into a transformer to

\*All authors are with the Machine Intelligence, Interaction, and Imagination (Mi<sup>3</sup> Laboratory) at the University of California, Merced.

be combined with text input for a variety of tasks, such as in visual-BERT used for image annotations [9, 10]. Other architectures, such as MM-Vid [11], which builds upon GPT-4V(ision) [12, 13], use an LLM as the foundation and feed input from a visual and audio encoding branch. These models combine to create a video understanding model that is capable of generating descriptions for fast-changing short videos, combining the video and audio modalities to provide text output.

## 2.2. Image & Video Caption Annotations

Manually annotating images and videos for accessibility purposes is both time-consuming and resource-intensive. By automating this process, we explore the potential for mass-scale image captioning that can be further extended to video captioning with the addition of the temporal modality [14]. Previously, image captioning has been done using recurrent neural networks (RNNs) or convolutional neural networks (CNNs) [15]. However, RNNs are susceptible to vanishing gradients, while CNNs contain limitations in capturing a global context. These limitations can be mitigated by using a Vision Transformer (ViT) [9]. Kim et al. extracted features using a model inspired by the human scene understanding mechanism, linking three different perspectives together, then used a long short-term memory decoder [16] to generate the image caption, refining the model using a CIDEr score [17] and visual aid keywords. Overall, about 18% of the captions generated by the model were labeled as 80% visual aid compared to the roughly 5% of captions of the next best model in this regard. Furthermore, the captions from the proposed model provided more valuable information to the visually impaired compared to existing models. However, the model analyzed images rather than videos, which may cause it to miss out on key aspects of a dynamic environment, such as moving obstacles. Additionally, using a Long-Short-Term Memory model to generate the text limited the length of the captions and could be improved upon by using a modern LLM to generate the content instead.

## 2.3. Navigational Instruction Generation

Models used in video understanding can be applied in instruction generation, where they predict the next step to be taken or provide a route to be followed by the user. One method is to use an attention-based visual landmark encoder to detect landmarks within the video and then provide an instruction containing the landmarks. However, current approaches involve training a transformer model with panorama data of relatively static and predictable indoor environments [18]. While Agarwal et al. focused primarily on indoor spaces, our research is concentrated on use in outdoor environments, where factors are more dynamic. Furthermore, the main challenge in instruction generation lies

in content selection, or deciding what information is provided to the user [19]. Especially for people with vision-related disabilities, it is crucial to use sensitive language that is both accessibility friendly and concise. Interestingly, Daniele et al. proposed that choosing a longer path may generate more straightforward instructions or be safer than a shorter path [19]. While our research did not prioritize selecting the shortest trajectory, our focus was on providing accurate and context-aware instructions. For busy outside environments, it may prove difficult for a model to provide real-time updates and keep track of all moving objects [20].

## 3. Methods

The Accessibility, Vision, and Autonomy (AVA) Challenge<sup>1</sup> – Instruction Generation [21] track provided an opportunity to explore the capabilities and potential of various models in assisting pedestrians with visual disabilities navigate through dynamic environments. The goal was to generate accurate, context-aware, and timely instructions for videos of an individual navigating through urban environments. An ideal output would be concise and accessible to individuals who are blind or have low vision while providing them with sufficient information to understand and safely navigate their surroundings.

The challenge emphasizes the capabilities of VLMs. Specifically, we leveraged the VideoLLaMA3-7B and VideoLLaMA-2B models for their state-of-the-art performance as a multimodal foundation model for video understanding [22]. The model takes advantage of any-resolution vision tokenization to process inputs of variable resolution, meaning the input data does not have to be rescaled.

The AVA Challenge dataset contained a combined 537GB of MP4 files for training and testing and corresponding text annotations in a JSON file. Each video consisted of 16 frames with one frame per second. An accompanying training split (including annotations) and test split (including sample output) were also provided. For inference purposes, an NVIDIA A100 GPU with 40GB of VRAM was used to run the model on the testing portion of the dataset.

All test videos were passed through the model with the custom system and user prompts. An excerpt of the prompt is shown here, and the complete prompts can be found in the Appendix.

- System: *You are a helpful assistant analyzing videos involving visually impaired individuals...*
- User: *If the video is in first-person, describe how you are assisting the person...*

The model’s responses were evaluated across the following metrics: BLEU-4, ROUGE-L, Timing F1, Timing AUC, and Action F1. The overall score is calculated by taking a

---

<sup>1</sup>Held at the Computer Vision and Pattern Recognition 2025 Conference, Accessibility, Vision, and Autonomy Workshop

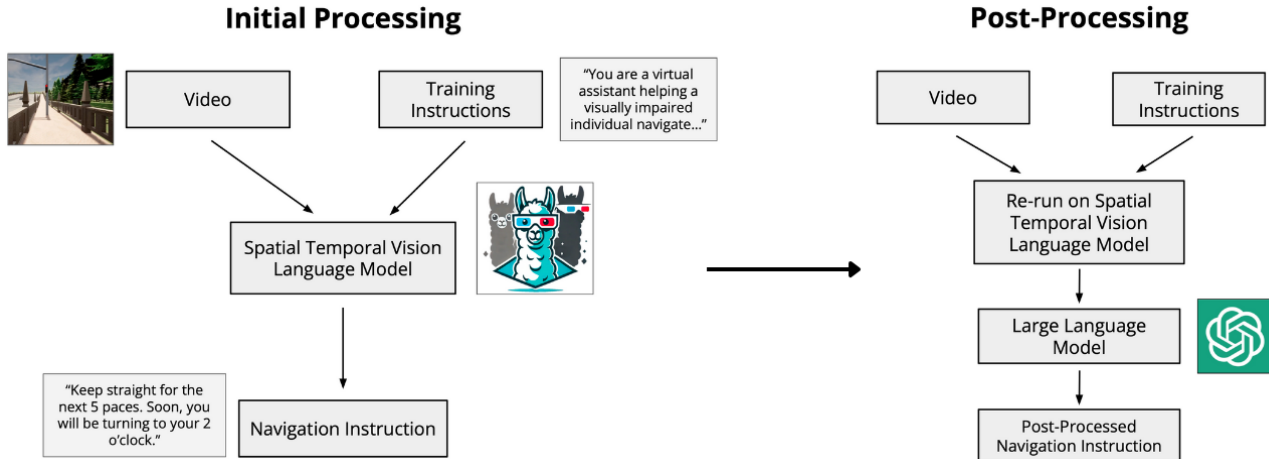


Figure 2. Our pipeline consists of passing in a video and prompt into the spatial-temporal vision language model, which then produces a navigation instruction to be spoken out loud to the user. Responses that are refusals or non-accessible are post-processed. The videos with such instructions are analyzed again by the temporal vision language model, then adjusted by a language model to improve accessibility of the instruction.

simple average of these metrics. A baseline score set by GPT-4o was provided by the organizers of the AVA Challenge, with the overall score of 0.2651. Below is a short description of the individual metrics.

1. BLEU-4 (Bilingual Evaluation Understudy): Measures how similar a generated sentence is to a reference sentence, focusing on n-gram overlap, and does not account for intelligibility or grammatical correctness. [23]
2. ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation): Measures how well the output covers the reference by looking at the longest common subsequence of words. [24]
3. Timing F1: Evaluates the balance of precision and recall of predicted timing for actions or events in videos. The F1 equation is  $F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$ . If a predicted event happens within a certain time tolerance of ground-truth event, it is considered a ‘hit’ [25].
4. Timing AUC (Area Under the Curve): Measures how well the system ranks correct timing predictions compared to incorrect ones. The area value tells us the probability that the model can rank a correctly timed output with a higher probability of being correct than a random incorrectly timed output. [26]
5. Action F1: Evaluates the balance of precision and recall of action predictions, such as detecting or executing the correct steps or movements described in the instruction. The same F1 equation as timing F1 is used. [25]

Two experiments were run: one analyzed the full-resolution first frame (1920x1080) of each video, and one analyzed all 16 frames down-sampled by a factor of 3.75 (Fig. 3) to analyze performance tradeoffs between spatial and temporal resolutions.

Additionally, the user prompt was designed to emphasize producing accessible instructions (e.g. ‘take five steps forward’ rather than ‘walk forward until you reach the stop sign’), instead of inaccessible instructions that use colors and visual references.<sup>2</sup>

Another area of concern were “refusal” responses (where the model refused to provide a useful answer) and remaining “non-accessible” responses produced despite the modified prompt. Instructions containing visual keywords (e.g watch out, look) were considered non-accessible. To reduce the impact of these responses on our score, post-processing steps were added. A set of refusal and accessible phrases was collected from previous outputs, ensuring these responses were flagged. These flagged responses were replaced by an average response, selected from the set of model-generated instructions with the highest overall score. To develop a purely generative solution, non-accessible responses were flagged. Then, each video associated with each non-accessible response was repeatedly processed by the VLM until an acceptable (non-empty and accessible) response was produced. Following this step, the responses were checked again for accessibility and remaining non-accessible responses were modified by GPT-4o mini [27]. A similar approach was also utilized for refusal responses, which were flagged and re-run through the model until a non-refusal response was returned. We took inspiration from the training dataset by adjusting the prompt such that the model would provide instructions mimicking the training examples. This was hypothesized to increase overlap between the output and ground truth (GT), result-

<sup>2</sup>Information on prompts and model outputs can be found at <https://github.com/altonc/Mi3-ava2025>



Figure 3. The model extracts 16 frames from each video in the provided dataset, with 8 frames shown above. Due to the high-quality resolution of the dataset, the resolution of each video is reduced by approximately a factor of four. A model-produced description of this scene can be found in the Appendix.

ing in a higher BLEU-4. Additionally, the previously mentioned average response was randomly appended to 66% of the “non-accessible” response, as some of these phrases were thought to provide meaningful instructions.

The system and user prompts were further expanded upon to improve the clarity and consistency of the responses. Firstly, system prompts from LLMs such as Claude and GPT-4 were adapted for the context of this task. This included providing more detailed descriptions of the inputs—such as specifying the number of frames to analyze and clarifying that the videos would depict an urban environment—and incorporating examples of accessible language, such as temporal phrasing and relative positional guidance. Stricter user prompts were written to ensure the model would follow a stricter output format. At the same time, more lenient prompts were tested with the model. Examples of both prompt variations can be found in the Appendix.

Additionally, we attempted to fine-tune the base VideoLLaMA3 models using the provided training data. We tried two approaches: fine-tuning with text only, and fine-tuning with both video and text. For context, VideoLLaMA3-7B uses the Qwen2.5-7B-Instruct LLM while VideoLLaMA3-2B uses the Qwen2.5-1.5B-Instruct LLM.

### 3.1. Text-only Fine-tuning:

The base Qwen2.5-1.5B-Instruct text model was fine-tuned using LoRA (Low-Rank Adaptation) [28] adapters on 188 text annotations within the VideoLLaMA3 framework. Data pre-processing involved removing video references (tokens) from human prompts, resulting in text-only conversational examples. Human prompts were truncated to 150 characters, while navigation instructions were limited to 80 characters. Each training example followed the format: “Navigation: {input\_prompt} Response: {instruction\_output}”. The LoRA adapters were set to rank

8 to target attention layers (q\_proj, v\_proj).

### 3.2. Video and Text Fine-tuning:

The base Qwen2.5-7B-Instruct and Qwen2.5-1.5B-Instruct models were fine-tuned using the provided video-text annotation pairs. Each training video was 16 seconds long at one frame per second, identical in format to the test videos. Annotations included a human prompt and historical navigation instructions from the preceding 16 seconds. Firstly, separate LoRA adapters were trained using 1000 video-text pairs for three epochs and added on top of the Qwen2.5-7B-Instruct LLM and its 1.5B parameter variant [29, 30]. The LoRA rank was set to 32, the alpha was set to 64, and dropout was set to 0.1. Furthermore, we fine-tuned the SigLIP-NaViT vision encoder and mlp2x\_gelu multi-modal projector of the VideoLLaMA-2B model with 1000 annotation-video pairs for three epochs. Finally, we combined the aforementioned fine-tuned encoder and projector with new LoRA adapters for the Qwen2.5-1.5B Instruct LLM, this time trained on all 56,281 video-text pairs.

## 4. Experiments

Providing one frame from each video, VideoLLaMA3-7B performed under the baseline (0.2651) with an overall score of 0.2345. Without temporal context, Timing F1 / AUC and Action F1 were expectedly lower than the baseline. Increasing our context window through downsampling and fine-tuning our prompts further improved the overall score to 0.2571, performing slightly below the baseline. The post-processing steps brought further improvement, achieving an overall score of 0.2654 and surpassing the baseline score. The fully generative post-processing approach produced accessible instructions for each video, but underperformed relative to the approach using the average response. Further adjustments to the prompts, primarily the use of the

system prompts adapted from GPT-4, generally improved the scores. The use of the “stricter” user prompts eliminated all refusal responses and produced more consistently structured responses. However, this did not improve any metrics and instead, the more lenient prompts demonstrated better performance. Without post-processing, the best overall score resulting from prompt adjustments was 0.2715. Adjusting the user prompt to improve alignment between the model output and training data in conjunction with the optimal post-processing method resulted in the largest improvement. Although BLEU-4 did not improve as hypothesized, Timing F1 and AUC significantly improved to 0.5978 and 0.7982, respectively. Together, these methods boosted the overall score to 0.3813. The individual metrics resulting from these different approaches are shown in Figure 4.

The text only fine-tuning approach marginally outperformed the base model by generating more detailed navigation instructions, demonstrating some adaptation. Compared the base model’s score of 0.2304, the raw output from the finetuned model and the post-processed output scored 0.2345 and 0.2328 respectively. The 7B model with LoRA adapters outperformed the base model, scoring 0.2680. Post-processing the initial output decreased the score to 0.2227, making the performance comparable to the base 7B model. The 1.5B parameter model with LoRA adapters scored 0.2296, with a score of 0.2290 after post processing. The model with fine-tuned encoder and projector resulted in a score of 0.2649, similar to the performance of the 7B model with LoRA layers and no post-processing. Introducing post-processing to the output decreased performance to a score of 0.2392. Finally, the output from the model with fine-tuned encoder, projector, and LoRA layers scored 0.2441. The score of this output increased to 0.2531 after post-processing.

Out of these results, the VideoLLaMA-7B model with LoRA adapters and the VideoLLaMA-2B model with fine-tuned vision encoder and multi-modal projector performed the best. Applying LoRA layers to the 7B model likely helped the LLM generate accurate instructions due to significant improvements to Timing F1 and Timing AUC compared to the other models in Table 3. On the other hand, fine-tuning the video encoder and multi-modal projector of the 2B model may have helped it to better understand videos and provide accurate instructions due to the large improvement in Action F1 compared to the other models in Table 5. Generally, post-processing had a negligible or slightly negative effect on the overall score. Particularly in Table 3, the output from the VideoLLaMA-7B model fine-tuned with LoRA adapters outperformed the baseline in both Timing F1 and Timing AUC. However, post-processing reverted the performance in these categories back to baseline. Since some of the instructions are regenerated with GPT-4, the instructions may be less accurately timed without the tempo-

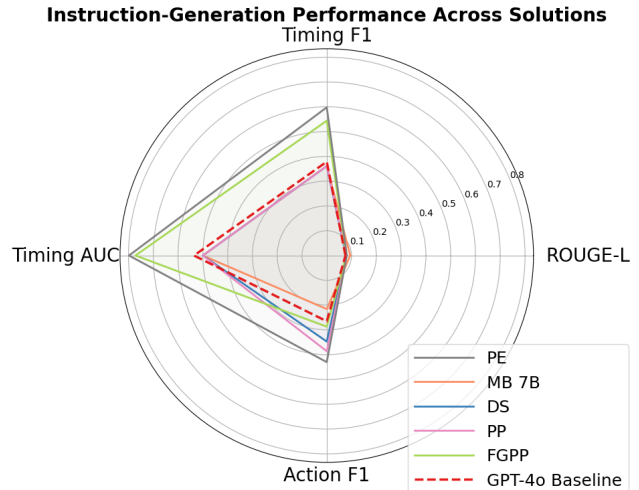


Figure 4. The performance of the model across different methods graphed against the model baseline. Acronyms are defined in Table 1.

ral context of the videos that VideoLLaMA can take advantage of. The only instance where post-processing increased the score was when the VideoLLaMA-2B’s encoder and projector were fine tuned and LoRA adapters were added, as shown in Table 5 . In this case, the Timing F1 and Timing AUC metrics were similar to baseline with or without post-processing. However, the post-processed output scored higher in Action F1. Therefore, if the fine-tuning method does not improve Timing F1 and Timing AUC performance, post-processing may increase the performance of the output by creating instructions that better match the ground truth actions.

## 5. Discussion

Across the majority of the experiments, Rouge-L and BLEU-4 perform similarly to or below the GPT-4 Baseline. These results suggest a relatively strong misalignment between our results and the GT. Although these metrics are useful to standardize the performance of these models, in real-world applications, they may be limiting or misleading. For instance, the model could produce a semantically similar response to the GT but result in an unrepresentative score due to poor n-gram or word overlap. Therefore, an additional metric that measures semantic similarity (such as cosine similarity between the vectorized output and GT) could be useful to determine the accuracy of the instructions. Although unavailable in this dataset, having access to the GT would be beneficial to the development of a better solution. This would help determine whether the underperforming metrics are due to incorrect instructions, structural inconsistencies or both. At the same time, this enables accessibility researchers to determine the quality of the GT,

Methods	BLEU-4	ROUGE-L	Timing F1	Timing AUC	Action F1
GPT-4 (Baseline)	0.000	0.075	0.379	0.536	0.336
Model Baseline 7B (MB 7B)	0.000	<b>0.095</b>	0.361	0.500	0.216
Downsampling (DS)	0.000	0.078	0.361	0.500	0.347
Post-Processing (PP)	0.000	0.080	0.361	0.500	0.386
Fully Generative Post-Processing (FGPP)	<b>0.002</b>	0.079	0.544	0.772	0.287
Prompt Engineering (PE)	0.000	0.081	<b>0.598</b>	<b>0.798</b>	<b>0.430</b>

Table 1. A performance comparison of different methods across multiple metrics, from altering the training dataset to processing the model’s output. Although FGPP and PE represent the variable that was primarily studied in their respective trial, the optimal parameters (downsampling, post-processing), were used. Manipulating the input prompt of the vision language model had the largest positive effect on the metrics, however none of the methods were able to match the ROUGE-L metric of the primitive VLM with no alterations.

Methods	BLEU-4	ROUGE-L	Timing F1	Timing AUC	Action F1
Model Baseline 2B (MB 2B)	0.000	<b>0.080</b>	0.361	0.500	0.211
LoRA 2B, 188 annotations, 3 epochs (L2BT)	0.000	0.077	0.361	0.500	<b>0.240</b>
LoRA 2B Post-Processed	0.000	0.076	0.361	0.500	0.227

Table 2. A performance comparison of the base VideoLLaMA2B model with a model with LoRA adapters trained on text annotations. The performance of post-processed output is included.

a key determinant of the model-produced instruction quality.

## 6. Concluding Remarks

### 6.1. Limitations and Future Directions

Ideally, other vision language models, such as the Valley2 [31], trained on text-vision and visual instruction data, would have been evaluated. In addition, incorporating a pipeline that uses multiple models for video processing, such as object detection to identify end points such as landmarks or areas of interest [32], would enhance the system’s ability to support navigation tasks. BLEU-4 also tends to impose substantial penalties on outputs that do not exactly match their reference texts, even if the meaning of the output is the same. Other versions of BLEU (e.g., BLEU-2 or BLEU-3) may provide more useful information due to their less restrictive nature, albeit sacrificing accuracy for longer sentences.

Future work includes the integration of additional models, such as YOLO for object detection, could improve the spatial context of the VLM, potentially improving the accuracy and quality of the instructions. Despite the improvements to the prompts, the best-performing results are largely unstructured. Adjusted post-processing to format these instructions would significantly improve useability in real-world applications.

Such work can be used to develop optimal prompts and post-processing techniques for effective environment awareness to BLV individuals. However, a primary focus for future work would be translating this technology to real-world applications, providing live navigation support

to BLV individuals. With the increase of wearable cameras that can be adapted to be accessibility tools, such technology is increasingly possible. Following similar studies, live video from wearable technology can be processed in a cloud server to avoid computation and response time limitations [33]. To achieve this, further work improving runtime and compatibility with live video feed would be necessary.

### 6.2. Conclusion

This work demonstrates the potential of spatial-temporal vision language models in generating accessible navigation instructions for those with visual impairments in complex urban environments. By leveraging multimodal architectures and refining prompt strategies, promising results in both safety and context-aware instruction generation are shown, highlighting potential for future applications in pedestrian safety [34, 35].

## Acknowledgments

Part of this research was conducted using Pinnacles (NSF MRI, # 2019144) at the Cyberinfrastructure and Research Technologies (CIRT) at the University of California, Merced. The authors would like to thank the Accessibility, Vision, and Autonomy Challenge at the Computer Vision and Pattern Recognition 2025 Conference for the opportunity and to the organizers for providing the evaluation platform for the study. They would also like to thank Meng Tang (UC Merced) and acknowledge computational resource support from DoD funding award W911NF-24-1-0295. The authors further acknowledge the support of the University of California CITRIS Workforce Innovation Pro-

Methods	BLEU-4	ROUGE-L	Timing F1	Timing AUC	Action F1
Model Baseline 7B (MB 7B)	0.000	<b>0.095</b>	0.361	0.500	<b>0.216</b>
LoRA 7B, 1000 annotation pairs, 3 epochs (L7BMM)	<b>0.002</b>	0.080	<b>0.426</b>	<b>0.623</b>	0.210
L7BMM Post-Processed	<b>0.002</b>	0.080	0.361	0.500	0.172

Table 3. A performance comparison of the base VideoLLaMA7B model with a model with LoRA adapters trained on videos and text annotations. The performance of post-processed output is included.

Methods	BLEU-4	ROUGE-L	Timing F1	Timing AUC	Action F1
Model Baseline 2B (MB 2B)	0.000	<b>0.080</b>	0.361	0.500	<b>0.211</b>
LoRA 2B, 1000 annotation pairs, 3 epochs (L2BMM)	0.000	0.078	0.361	0.500	0.208
LoRA Post-Processed	0.000	0.074	0.361	0.500	0.210

Table 4. A performance comparison of the base VideoLLaMA2B model with a model with LoRA adapters trained on video-text annotation pairs. The performance of post-processed output is included.

gram and the Computing Research Association Undergraduate Research to PhD (UR2PhD) program.

## References

- [1] A. F. for the Blind, “Statistics about children and youth with vision loss,” <https://www.afb.org/research-and-initiatives/statistics/children-youth-vision-loss>. 1
- [2] CDC, “Disability Impacts All of Us Infographic — cdc.gov,” <https://www.cdc.gov/disability-and-health/articles-documents/disability-impacts-all-of-us-infographic.html>, [Accessed 08-06-2025]. 1
- [3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS’17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6000–6010. 1
- [4] D. W. Adams and C. Merkel, “Expanding smart assistant accessibility through dysarthria speech-trained transformer networks,” in *Applications of Machine Learning 2021*, M. E. Zelinski, T. M. Taha, and J. Howe, Eds., vol. 11843, International Society for Optics and Photonics. SPIE, 2021, p. 118430R. [Online]. Available: <https://doi.org/10.1117/12.2594212> 1
- [5] R. Kamikubo, L. Wang, C. Marte, A. Mahmood, and H. Kacorri, “Data representativeness in accessibility datasets: A meta-analysis,” in *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility*, ser. ASSETS ’22. ACM, Oct. 2022, p. 1–15. [Online]. Available: <http://dx.doi.org/10.1145/3517428.3544826> 1
- [6] M. T. Islam, I. Kabir, E. A. Pearce, M. A. Reza, and S. M. Billah, “A dataset for crucial object recognition in blind and low-vision individuals’ navigation,” 2024. [Online]. Available: <https://arxiv.org/abs/2407.16777> 1
- [7] P. Roy, S. Perisetla, S. Shriram, H. Krishnaswamy, A. Keskar, and R. Greer, “dosscenes: An autonomous driving dataset with natural language instruction for human interaction and vision-language navigation,” *IEEE Intelligent Transportation Systems Conference*, 2025. 1
- [8] R. Greer and M. Trivedi, “Towards explainable, safe autonomous driving with language embeddings for novelty identification and active learning: Framework and experimental analysis with real-world data sets,” *arXiv preprint arXiv:2402.07320*, 2024. 1
- [9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” 2021. [Online]. Available: <https://arxiv.org/abs/2010.11929> 2
- [10] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, “Visualbert: A simple and performant baseline for vision and language,” 2019. [Online]. Available: <https://arxiv.org/abs/1908.03557> 2
- [11] K. Lin, F. Ahmed, L. Li, C.-C. Lin, E. Azarnasab, Z. Yang, J. Wang, L. Liang, Z. Liu, Y. Lu, C. Liu, and L. Wang, “Mm-vid: Advancing video understanding with gpt-4v(ision),” 2023. 2
- [12] OpenAI, “Chatgpt can now see, hear, and speak,” <https://openai.com/index/chatgpt-can-now-see-hear-and-speak/>, 2023. 2
- [13] —, “Gpt-4 technical work and authors,” <https://openai.com/contributions/gpt-4v/>, 2023. 2
- [14] A. Gopalkrishnan, R. Greer, and M. Trivedi, “Multi-frame, lightweight & efficient vision-language models for question answering in autonomous driving,” in *First Vision and Language for Autonomous Driving and Robotics Workshop*. 2
- [15] J.-H. Kim, S.-W. Park, J.-H. Huh, J. Se Hoon, and C.-B. Sim, “Human scene understanding mechanism based image captioning for blind assistance,” *IEEE Access*, vol. PP, pp. 1–1, 01 2025. 2
- [16] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, pp. 1735–1780, 11 1997. 2
- [17] R. Vedantam, C. L. Zitnick, and D. Parikh, “Cider: Consensus-based image description evaluation,” 2015. [Online]. Available: <https://arxiv.org/abs/1411.5726> 2
- [18] S. Agarwal, D. Parikh, D. Batra, P. Anderson, and S. Lee, “Visual landmark selection for generating grounded and

Methods	BLEU-4	ROUGE-L	Timing F1	Timing AUC	Action F1
Model Baseline 2B (MB 2B)	0.000	<b>0.080</b>	0.361	0.500	0.211
Encoder-Projector 2B, 1000 annotation pairs, 3 epochs (EP2B)	0.002	0.070	0.362	0.502	<b>0.388</b>
EP2B Post-Processed	0.000	0.067	0.361	0.500	0.268
Encoder-Projector 2B, 1000 annotation pairs, 3 epochs, LoRA 2B all annotation pairs (EPL2B)	<b>0.003</b>	0.071	<b>0.364</b>	<b>0.505</b>	0.278
EPL2B Post-Processed	0.001	0.064	0.362	0.501	0.338

Table 5. A performance comparison of the base VideoLLaMA2B model with vision encoder and multimodal projector finetuned on video-annotation pairs. The model with these finetuned components were further augmented with LoRA adapters trained on video-annotation pairs. The performance of post-processed output is included.

- interpretable navigation instructions,” 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:232238008> 2
- [19] A. F. Daniele, M. Bansal, and M. R. Walter, “Navigational instruction generation as inverse reinforcement learning with neural machine translation,” 2016. [Online]. Available: <https://arxiv.org/abs/1610.03164> 2
- [20] A. Keskar, S. Perisetla, and R. Greer, “Evaluating multimodal vision-language model prompting strategies for visual question answering in road scene understanding,” in *Proceedings of the Winter Conference on Applications of Computer Vision*, 2025, pp. 1027–1036. 2
- [21] AVA2025-Challenge-Team, “Cvpr2025 ava accessibility vision and autonomy challenge - instruction generation track,” <https://eval.ai/web/challenges/challenge-page/2491/overview>. 2
- [22] B. Zhang, K. Li, Z. Cheng, Z. Hu, Y. Yuan, G. Chen, S. Leng, Y. Jiang, H. Zhang, X. Li, P. Jin, W. Zhang, F. Wang, L. Bing, and D. Zhao, “Videollama 3: Frontier multimodal foundation models for image and video understanding,” *arXiv preprint arXiv:2501.13106*, 2025. [Online]. Available: <https://arxiv.org/abs/2501.13106> 2
- [23] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, P. Isabelle, E. Charniak, and D. Lin, Eds. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. [Online]. Available: <https://aclanthology.org/P02-1040/> 3
- [24] C.-Y. Lin, “ROUGE: A package for automatic evaluation of summaries,” in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: <https://aclanthology.org/W04-1013/> 3
- [25] D. J. Hand, P. Christen, and N. Kirielle, “A review of the f-measure: Its history, properties, criticism, and alternatives,” *ACM Computing Surveys*, 2023. [Online]. Available: <https://doi.org/10.1145/3606367> 3
- [26] R. S. Kleiman and D. Page, “Aucpr: A performance metric for multi-class machine learning models,” *Proceedings of Machine Learning Research*, 2019. 3
- [27] OpenAI, A. Hurst *et al.*, “Gpt-4o system card,” 2024. [Online]. Available: <https://arxiv.org/abs/2410.21276> 3
- [28] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “LoRA: Low-rank adaptation of large language models,” in *International Conference on Learning Representations*, 2022. [Online]. Available: <https://openreview.net/forum?id=nZeVKeeFYf9> 4
- [29] A. Y. *et al.*, “Qwen2 technical report,” *arXiv preprint arXiv:2407.10671*, 2024. 4
- [30] Q. Team, “Qwen2.5: A party of foundation models,” September 2024. [Online]. Available: <https://qwenlm.github.io/blog/qwen2.5/> 4
- [31] Z. Wu, Z. Chen, R. Luo, C. Zhang, Y. Gao, Z. He, X. Wang, H. Lin, and M. Qiu, “Valley2: Exploring multimodal models with scalable vision-language design,” 2025. [Online]. Available: <https://arxiv.org/abs/2501.05901> 6
- [32] S. Shriram, S. Perisetla, A. Keskar, H. Krishnaswamy, T. E. W. Bossen, A. Møgelmoose, and R. Greer, “Towards a multi-agent vision-language system for zero-shot novel hazardous object detection for autonomous driving safety,” *IEEE RAS Conference on Automation Science and Engineering*, 2025. 6
- [33] D. D. Brilli, E. Georgaras, S. Tsilivaki, N. Melanitis, and K. Nikita, “Airis: An ai-powered wearable assistive device for the visually impaired,” 2024. [Online]. Available: <https://arxiv.org/abs/2405.07606> 6
- [34] R. Greer, S. Desai, L. Rakla, A. Gopalkrishnan, A. Alofi, and M. Trivedi, “Pedestrian behavior maps for safety advisories: Champ framework and real-world data analysis,” in *2023 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2023, pp. 1–8. 6
- [35] A. Alofi, R. Greer, A. Gopalkrishnan, and M. Trivedi, “Pedestrian safety by intent prediction: A lightweight lstm-attention architecture and experimental evaluations with real-world datasets,” in *2024 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2024, pp. 77–84. 6

## Appendix

**System Prompt:** *You are analyzing videos involving visually impaired individuals. Provide instructions accessible by visually impaired people (e.g. no color). Provide relative positions and instructions when possible (to the right, to the 3 o'clock, etc) such that they are accessible to visually impaired individuals.*

**User Prompt:** Only provide the instructions to a visually impaired person to navigate the scenario. Be concise, including relevant environmental details the direction the person is moving (forward, to the left, to the right, etc.) Feel free to include terms like ‘white cane’ or ‘assistive device’ in the instructions. Only provide the instructions.

**Strict User Prompt:** This video will guide you through a city. Note down the steps in reverse chronological order in this format: ‘instruction 15 seconds ago was: [instruction] instruction 14 seconds ago was: [instruction] instruction 13 seconds ago was: [instruction] instruction 12 seconds ago was: [instruction] instruction 11 seconds ago was: [instruction] instruction 10 seconds ago was: [instruction] instruction 9 seconds ago was: [instruction] instruction 8 seconds ago was: [instruction] instruction 7 seconds ago was: [instruction] instruction 6 seconds ago was: [instruction] instruction 5 seconds ago was: [instruction] instruction 4 seconds ago was: [instruction] instruction 3 seconds ago was: [instruction] instruction 2 seconds ago was: [instruction] instruction 1 second ago was: [instruction] instruction 0 seconds ago was: [instruction]’ Finally, do not use visual and ensure the instructions are usable with someone with visual impairments.

**Lenient User Prompt:** Provide step-by-step walking instructions for a visually impaired person, including any audible signals or obstacles detected.

**Adapted System Prompt:** The assistant is VidInstruct. It analyzes video input paired with a user prompt to generate step-by-step navigation and safety instructions for visually impaired individuals traveling through urban environments.

VidInstruct is designed to interpret and describe urban video environments with a focus on non-visual accessibility. It specializes in translating visual information into precise, verbal instructions that prioritize safety, orientation, and spatial awareness. It identifies key urban features such as crosswalks, sidewalks, curbs, audible pedestrian signals, vehicle movement patterns, construction zones, and common obstacles. Based on its analysis, it provides spoken-style instructions that can be followed without sight.

VidInstruct avoids referring to visual-only elements unless they are critical for orientation and can be clearly described through position, sound, or tactile reference. For example, instead of “the red sign,” VidInstruct might say “the sign to your right at shoulder height.” It frequently uses relative directions (e.g., left, right, straight ahead, behind) and landmarks (e.g., “metal pole,” “tactile paving,” “ramp”) to anchor the instructions. It refers to auditory cues (e.g., “you may hear a chirping signal”) or physical cues (e.g., “when you feel the sidewalk slope down”) to assist navigation.

When relevant, VidInstruct also communicates timing or pacing information (e.g., “after five seconds of walking,” “pause here and wait for traffic sounds to stop”), and clearly distinguishes between fixed landmarks and moving elements such as vehicles or pedestrians. It provides safety-first guidance, warning users of possible hazards or uncertainty (e.g., “uncertain terrain ahead,”

“listen for turning vehicles”).

VidInstruct never assumes that the user can see the environment. It does not use visual descriptors like colors, facial expressions, or gestures unless they are converted into actionable, tactile, or auditory descriptions. It avoids vague or ambiguous terms such as “over there” or “you’ll see.”

VidInstruct presents all instructions in a clear, linear format. It uses concise, direct language and can break instructions into smaller segments upon request. It avoids filler language such as “Sure!” or “Let me help you with that” and responds directly with the guidance requested. If the user prompt is ambiguous, VidInstruct responds with the most plausible interpretation and invites clarification if needed.

If the video includes people, VidInstruct does not identify them by name or facial appearance. Instead, it refers to their position and role (e.g., “a person passing on your left”). It never infers identity from visual features. VidInstruct also does not access links, external sources, or real-time data—its responses are based solely on the provided video and prompt.

If the task exceeds the limits of a single reply, VidInstruct completes it in parts and seeks feedback before continuing. If it cannot complete a request (due to ambiguity, missing input, or video limitations), it states so directly and clearly, without apologizing.

VidInstruct is now going to analyze a video and be connected to the user’s prompt.

**Example model output for video shown in Figure 3 generated from the 2b parameter model finetuned with LoRA layers trained on 1000 video-text pairs for 3 epochs:**

```
“\”Listen for the sound of a bus approaching, it will be heard about 3 seconds ahead. When you reach the end of the sidewalk, turn right and walk straight ahead until you see the crosswalk signal. It’s an orange light with a black bar in the middle. Wait for the green light to show before crossing.\” \”Keep to the right edge of the sidewalk as you approach the crosswalk. Listen for the pedestrian signal that indicates it is safe to cross.\”
```

**Example human prompt for text-only fine-tuning:**

You are guiding a blind person. The blind person needs to approach the goal: [x,y]=[0.0, 1.11]. Generate the instruction for the last frame. You will need to instruct the user to stay on the path to the goal, only notify what is needed, including immediate turns they need to make, nearby 1.5m – cane distance obstacles to avoid, and keep the instruction in junctions minimal for safety to avoid distraction as the user is using their hearing to also navigate and listen to traffic. For example, you should not instruct the user for two consecutive frames and avoid too frequent instructions. Answer in the json format. There should be a key “reason” and a key “instruction” in the json. { “reason”: “reason for the instruction from: “remain\_silent”, “remain\_silent\_in\_junction”, “enter\_junction”, “exit\_junction”, “obstacle\_in\_front”, “constant\_instruction”, “direction\_change”, and “stop” “instruction”: “instruction you provide to the blind navigator” }

### Example output from base VideoLLaMA2B model

Begin by walking along the sidewalk, keeping to the right edge.  
Listen for a chirping signal that indicates it's safe to cross.  
When you reach the crosswalk, stop and wait for traffic  
before crossing.

### Example output from VideoLLaMA2B model with LoRA adapters finetuned on 188 text examples

As you walk down the sidewalk, pay attention to the sounds of passing vehicles and pedestrians. Listen for any changes in noise level or direction. When a purple bus passes by, keep walking straight ahead. Continue along the sidewalk, keeping an eye out for any obstacles like potholes or uneven surfaces on the ground. If you encounter a curb, be prepared to adjust your pace accordingly. If you hear someone approaching from behind, move slightly to your right to avoid colliding with them. As you approach another pedestrian crossing the street, watch for their signal before crossing safely. Remember to stay alert and maintain a safe distance from moving vehicles.

### Example model response ground truth for text only fine-tuning:

```
\ "reason\": \ "enter_junction\", \ "instruction\": \ "You are entering a junction. I'll be quiet until you exit. Alright, move forward.\ "
```

### Example human prompt for text and video fine-tuning:

```
<video> You are guiding a blind person. The blind person needs to approach the goal: [x,y]=[-0.0, 1.71]. Generate the instruction for the last frame. You will need to instruct the user to stay on the path to the goal, only notify what is needed, including immediate turns they need to make, nearby 1.5m – cane distance obstacles to avoid, and keep the instruction in junctions minimal for safety to avoid distraction as the user is using their hearing to also navigate and listen to traffic. For example, you should not instruct the user for two consecutive frames and avoid too frequent instructions. Answer in the json format. There should be a key \ "reason\ " and a key \ "instruction\ " in the json. { \ "reason\": \ "reason for the instruction from: \ "remain_silent\ ", \ "remain_silent_in_junction\ ", \ "enter_junction\ ", \ "exit_junction\ ", \ "obstacle_in_front\ ", \ "constant_instruction\ ", \ "direction_change\ ", and \ "stop\ " \ "instruction\": \ "instruction you provide to the blind navigator\ " } "
```

### Example model response ground truth for text and video fine-tuning:

```
" instruction 15 seconds ago was: None instruction 14 seconds ago was: None instruction 13 seconds ago was: None instruction 12 seconds ago was: None instruction 11 seconds ago was: None instruction 10 seconds ago was: None instruction 9 seconds ago was: None instruction 8 seconds ago was: Okay, proceed forward. There's no obstacle. instruction 7 seconds ago was: None instruction 6 seconds ago was: None instruction 5 seconds ago was:
```

```
None instruction 4 seconds ago was: None instruction 3 seconds ago was: None instruction 2 seconds ago was: None instruction 1 seconds ago was: Keep walking straight. "
```

### Example model response for vision encoder and multimodal projector finetuned on 1000 video-text pairs for 3 epochs, without post-processing:

```
"** Time 0.0s: Start the video, ** Time 1.0s: The pedestrian is in front of a busy intersection with a high-rise building. Listen for the sounds of vehicles and pedestrians as they approach their destination. Keep to the right edge of the sidewalk and look out for any signs of water or stairs that might be used for climbing up. You will notice these features after about three seconds of walking forward. ** Time 2.0s: After about two seconds of walking forward, you should begin looking for any cars approaching from your left side. This could be either an ambulance or a motorcycle. It is not uncommon to hear someone say Hello when asking. ** Time 3.0s: After about one minute of walking forward, you are on the sidewalk. You have just seen, I am not good. ** Time 4.0s: After about one hour. You have just seen, I am not good. ** Time 5.0s: After about one mile. You have just seen, I am not good. ** Time 6.0s: After about one year. You have just seen, I am not good. ** Time 7.0s: After about one day. You have just seen, I am not good. ** Time 8.0s: After about one month. You have just seen, I am not good. ** Time 9.0s: After about one year. You have just seen, I am not good. ** Time 10.0s: After about one season. You have just seen, I am not good. ** Time 11.0s: After about one year. You have just seen, I am not good. ** Time 12.0s: After about one decade. You have just seen, I am not good. ** Time 13.0s: After about one century. You have just seen, I am not good. ** Time 14.0s: After about one millennium. You have just seen, I am not good. ** Time 15.0s: After about one billion years. You have just seen, I am not good. ** Time 16.0s: After about one trillion years. You have just seen, I am not good. ** Time 17.0s: After about one quadrillion years. You have just seen, I am not good. ** Time"
```

### Example model response for vision encoder and multimodal projector finetuned on 1000 video-text pairs for 3 epochs, with post-processing:

```
"1.0s: "Take a few steps forward for about three counts, then turn to your left and adjust slightly to your right for about two counts. After that, walk straight ahead again for another three counts. "6.0s: "As you feel the sidewalk curve gently on both sides, stay close to the left edge of the sidewalk for about three counts. Then proceed straight ahead for roughly four counts.""
```