

SPIE Proceedings: Transformer-Based Image Captioning as a Framework for Defense Applications

Devin J Ullerick^a, Dzmitry Kasinets^{a,c}, Jayeeta Ghosh PhD^b, Dilshad Akkam Veettil PhD^b, Amir K. Saeed^a, Benjamin A. Johnson^a, and Benjamin M. Rodriguez, PhD^a

^aJohns Hopkins University Whiting School of Engineering, Baltimore, MD 21218, USA

^bAWS Professional Services, Seattle, Washington 98170, USA

^cUniversity of Florida Gainesville, Florida, USA

ABSTRACT

Transformer models have revolutionized the field of image captioning, offering advanced capabilities through self-attention mechanisms that capture intricate visual and textual relationships. This paper presents an innovative approach to applying transformer models for image captioning. Current State-of-the-Art (SOTA) performance has only been achieved by large vision-language models (LVLMs). Our approach distills the capabilities of a large model into a smaller model, thereby preserving much of its capabilities while being deployable on edge devices for image captioning in a live environment. A proof-of-concept (POC) system uses the German Street Signs dataset from Kaggle as a controlled test case. This experiment investigates the improvement in visual reasoning and decision-making from fine-tuning a small VLM using captions generated by a larger model. Experimental results demonstrate the distilled model has significantly improved visual reasoning, leading to more reliable and context-aware image captions. Our system can be extended to defense applications, where a distilled model can enhance the decision-making and operational effectiveness in a variety of environments.

Keywords: transformer models, vision-language models (VLMs), fine-tuning, image captioning, decision-making

1. INTRODUCTION

The course of development in the field of Artificial Intelligence (AI) has been ever-evolving over the past few years. The usage rate of Generative AI tools started accelerating in late 2022 with the release of OpenAI's ChatGPT.¹ Since then, OpenAI, Anthropic, Alibaba, Salesforce, Meta, and other AI labs have developed larger and larger Transformer-based models. During these years of exponential development, new kinds of AI model started being released that could work with multimodal data. These models became known as Vision-Language Models (VLMs), a counterpart to Large Language Models (LLMs) that were historically only capable of text in and text out. As of 2024 however, even VLMs began to expand to the point that researchers began referring to them as Large Vision-Language Models (LVLMs). Such models have the impressive ability to perform visual reasoning, instruction-following, decision-making, and other agentic tasks. However, deploying such a model on a small device, such as a smartphone or other edge device is impractical given the memory requirements of these models.

Storing such a LVLM on a small edge device is only feasible for the less performant smaller versions of these larger models. To address this limitation, researchers began to explore distilling large models into relatively smaller ones. Thereby, the capabilities of the smaller model could be retained, but still be hosted on such devices with less memory. Our study focuses on exactly this area. We explore the feasibility of teaching a smaller model to be nearly as performant as a State-of-the-Art (SOTA) model.

In this paper, we prove that distilling a larger VLM into a smaller VLM is possible for a specific image captioning task. Our experiments show that the abilities of two VLMs, when merged through distillation, can result in emergent behavior that rivals the capabilities of the larger model in some test cases.

In other studies, researchers have focused on distilling the self-attention mechanism of a large "Teacher" model through a process of taking its final Transformer layer and aligning the smaller "Student" model's output

to it. Our process differs from this approach by instead fine-tuning the Student model using the outputs of the Teacher model on a given task.

The remainder of this paper is structured as follows. In Section 2, we introduce previous work that is relevant to our study. Section 3 describes the methodological approach we took to develop our fine-tuned model and score its alignment to the Teacher model. We then present our final scores and other results in Section 4. Finally, we conclude with a brief summary of our findings and discuss opportunities for future work.

2. BACKGROUND

2.1 Vision-Language Models (VLMs)

Vision Language Models (VLMs)² represent a unique fusion of computer vision and natural language processing, that can learn from images and texts simultaneously. In most cases the models consist of an image encoder, an embedding projector to align image and text embeddings, and a text decoder for generation. There are different combinations of individual building blocks and modeling strategy to pretrain and finetune VLMs.

Open source VLMs, for example LLaVA,³ CogVLM,⁴ InstructBLIP,⁵ Qwen-VL-7B,⁶ and MiniGPT-4⁷ made significant improvements in multimodal models that can be applied for various downstream tasks like chatting about images, visual question and answering, image captioning, and complex visual reasoning.

As part of this research, we focused on exploring Vision Language Models (VLMs) with a relatively smaller computational footprint, tailored to our specific use case in defense applications. Our primary goal was to develop a system capable of generating accurate instructions for drivers navigating unfamiliar territories, often dealing with low-resolution imagery. InstructBLIP was particularly interesting as it is using Vicunna-7B as language model and works well for specific tasks. On the other hand, Qwen2-VL-7B Instruct model achieves SOTA performance on different image resolution and ratio and works well with multi-lingual texts.

2.2 Large vs Small Models

As Large Language Models (LLMs) are trained on increasingly vast datasets using longer chain-of-thought (CoT) reasoning and more advanced GPUs, the cost and computational demands required to run them have grown substantially. Like many technologies, later adopters benefit from the advancements and learnings of early pioneers, often achieving similar outcomes at reduced cost. In defense applications, where deployment often involves smaller, edge devices in the field, this becomes especially important. We explored whether the capabilities of a larger model could be fine-tuned and distilled into a smaller model while still achieving significant accuracy gains.

Recent research has highlighted tradeoffs and advantages in distilling and fine-tuning large models into smaller ones, particularly with respect to CoT reasoning depth.⁸ Our work focuses specifically on the fine-tuning aspect of this process, aiming to reduce model size while preserving performance. This has direct implications for defense use cases, where resource-efficient models must operate reliably under constrained computational and power conditions at the edge.

2.3 LLM-as-a-Judge

You have a 2 papers here on adjudication⁹ and LLM-as-a-judge¹⁰ Using LLMs as judges to evaluate other LLM-generated text is promising for the following reasons:

- Traditional metrics like distance scores often fail to capture true semantic similarity.
- Traditional metrics often misrepresent performance, as semantically similar words can still have large difference scores.
- Human evaluation is accurate but can be slow, tedious, and expensive — especially at scale, even with tools like Amazon Mechanical Turk.

LLM-as-a-Judge offers a scalable and more interpretable alternative for assessing text similarity. LLM-as-a-judge is useful for automated labeling, self-improvement, and task-specific ranking of LLMs. Judge LLMs can choose the correct answer from options (e.g., in math problems). For tasks like mathematical reasoning, they demonstrate complex multi-step reasoning. Larger models are typically more accurate judges due to their stronger multi-step reasoning abilities.⁹

2.4 Dataset

Talk about data distribution here.

3. METHODOLOGY

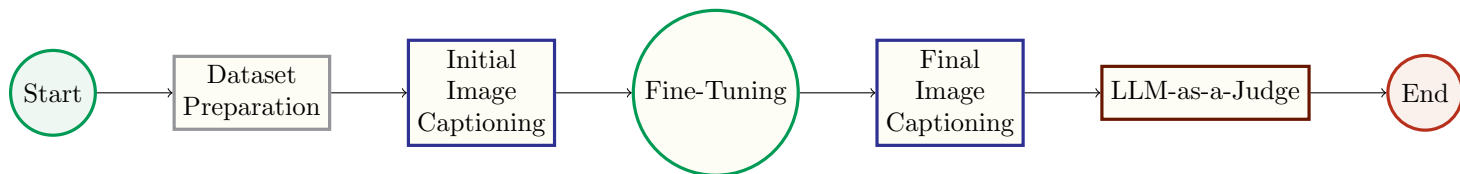


Figure 1. Methodology Overview

3.1 Dataset Preparation

For our image captioning system, the Kaggle dataset Traffic Signs Dataset¹¹ required some preprocessing. Our preprocessing pipeline cropped the signs from each image and saved the cropped and original images into cloud storage.

First, the full dataset was loaded from Kaggle via the KaggleHub API. Within the full dataset are JPEG and TXT files. The text files contain YOLO¹² format pixel coordinates for each of the street signs in the JPEG images. Looping through each text file, we calculated the bounding boxes for each street sign and then cropped it using the PIL python library.¹³ Once cropped, we then saved the original image and each of the cropped street signs in their own directory within the larger street sign directory. Structuring the parent directory in this way allowed us to later store text files containing the image captions generated by the VLMs, as well as other meta data used in our analysis. Once organized, we started our initial test of the image captioning ability for each candidate VLM.

3.2 Initial Image Captioning

To develop a system that is capable of distilling a larger model’s visual reasoning abilities into a smaller model, we first had to determine candidates for both the Teacher (larger) and Student (smaller) models. Potential Student models were provided a single cropped yield sign from one of the images in the German Street Signs dataset, shown in Figure 2.

The prompt used across all candidates was: "I have provided an example of a street sign from somewhere in Germany while I am driving my car. Provide a succinct explanation of what this traffic sign means and how I should react to it."

The captions shown in Table 1 for these sample images exposed the initial visual reasoning abilities of each of our models. For the Student model, we considered InstructBLIP Vicuna-7B⁵ and Qwen2-VL-7B-Instruct.⁶ We additionally tested the capabilities of ChatGPT’s o1 model¹⁴ because suspected that this model would provide the best captions. Therefore, it would serve as a good basis for choosing which Student model candidate came closest to its captioning ability.

InstructBLIP was quickly ruled out from consideration as it incorrectly captioned the yield sign as a pedestrian crossing sign. It is conceivable that this model interpreted the yield sign as a pedestrian crossing because of their similarities. Regardless, as this model failed in a single simple test case, we decided not to pursue it as our Student model.



Figure 2. Full image from the Kaggle dataset showing how sign cropped sign with bounding boxes.

VLM Image Captions	
Model Name	Caption
Instruct-BLIP Vicuna-7B	This traffic sign indicates that there is a pedestrian crossing ahead. As a driver, you should slow down and be prepared to stop if necessary to allow pedestrians to cross the road safely.
Qwen2-VL-7B-Instruct	As a driver, you should stop at the sign and give way to any vehicles or pedestrians that are already on the road. You should also look both ways to ensure that it is safe to proceed before continuing your journey. If there is no other traffic or pedestrians on the road, you may continue driving, but you should still be cautious and attentive to any potential hazards.
OpenAI o1	This is a "yield" sign ("Vorfahrt gewähren") commonly found in Germany. It indicates that you must give priority to other vehicles at the intersection or merge point ahead. How to react: Slow down as you approach the sign. Check for oncoming traffic on the priority road. Proceed only when it is safe and there is no risk of collision.

Table 1. Initial Test Image Captions for 2 2.

On the other hand, Qwen2-VL-7B-Instruct provided a correct, albeit lengthy, caption for the yield sign. Ideally, we want the caption to be succinct enough so that the driver does not require significant mental effort to interpret and act upon the instruction. However, we believed that through fine-tuning we could teach the model to provide shorter instructions.

Finally, we used OpenAI's o1 model as a test to see how much better its description would be than either Qwen or InstructBLIP. Unsurprisingly, o1 displayed the best image captioning ability. o1 had the most direct and clear instruction; additionally o1 impressively provides the German name for giving way ahead.

With these results, we chose to use Qwen2-VL-7B-Instruct model as our Student. We chose this model because of its success in captioning such a simple image and it is small enough to run on an edge device, such as the Jetson Orin.¹⁵ Such a capability would allow for future work deploying our final model onto this kind of edge device and outputting auditory instruction to the driver based on a camera-provided video feed.

Having chosen a Student model, we advanced to creating captions using Qwen2-VL-7B-Instruct across all 900 German street sign images. Our approach involved prompting the model with not only the full-scope picture, as in Figure 3 below, but also the individual cropped street signs contained within that image. Knowing how complex of an environment a roadway can be, we hoped that any additional context we could provide the model would aid its understanding and thereby its ability to provide useful instructions. Our hypothesis was that by providing both the original and cropped sign images, the self-attention mechanism of the model would pick up on the importance of the cropped signs in the full image. However, initial spot-checking of some of the generated captions showed that the model was still providing overly verbose instructions. We tested using both the 7B and 2B sized models, but neither provided an easily interpretable caption for a driver.



Figure 3. Image 00867 from the German street signs dataset.

To improve upon the initial captions supplied by Qwen, we next sought an appropriate Teacher model that could give more succinct directions. Initially, we explored creating a custom GPT in ChatGPT’s interface. We hoped to create a knowledge base of a set of images and a concise reference caption. Any captions generated by this custom GPT would hopefully follow the pattern of being concise. In our initial tests, this approach worked well. However, trying to have the custom GPT process an entire zip folder of image sets proved to be too difficult. Instead of providing simple captions with directions, as was the case during testing, the GPT would try to process the full folder at once and ended up continuously defaulting to using OCR, rather than any of its vision-language capabilities. The GPT would then apply rule-based directions based on the text extracted via OCR. The results were too simple and did not align with the knowledge base. However, this test was helpful in showing us how effective ChatGPT’s closed-source models are at visual reasoning.

Knowing that closed-source models typically provided correct and concise captions led us down the path of using such a model as our Teacher model. For the sake of frugal experimentation, we tested the efficacy of ChatGPT’s GPT-4o mini model to caption our German street sign image sets. Researchers believe that GPT-4o mini contains at least 8B parameters.¹⁶ It should be noted that a key difference of this model is that it requires encoding images into a UTF-8 format before processing, while the Qwen2 model does not tokenize images before passing them in the prompt. Despite this potential limitation, our initial spot checks found that GPT-4o mini’s captions were significantly better and more concise than both the 2B and 7B Qwen models’ captions. The captions for Figure 2 and its cropped street sign showing that there is an upcoming traffic light show the comparative abilities of these models is shown in Table 2

VLM Image Captions	
Model Name	Caption
Qwen2-VL-2B-Instruct	If you are driving down a street in Germany and come across these street signs, you should proceed with caution and follow the instructions provided by the signs. These signs are typically used to indicate specific traffic rules or regulations that apply to the area. It is important to pay attention to these signs and follow the instructions to ensure your safety and the safety of others on the road.
Qwen2-VL-7B-Instruct	next stop.
GPT-4o Mini	As you drive down the street, be aware of the traffic signals ahead and prepare to slow down or stop if necessary due to ongoing road work.

Table 2. "Teacher" Model testing for caption accuracy of Figure 2

Not only was the GPT-4o mini caption the only correct interpretation of the street sign, it was also concise enough for a driver to effectively act on it. Unfortunately, the Qwen2-2B model did not address the sign and instead gave a boilerplate cautionary caption. Despite the conciseness of the 7B model, the instruction of "next stop" neither addresses the sign nor is accurate given the absence of a visible traffic light.

Overall, the other GPT-4o mini captions tended to follow this pattern of being both correct and concise, which is exactly what we hoped to "teach" the Qwen model through fine-tuning.

3.3 Fine-Tuning

Based on our research, one of our best options for fine-tuning the Qwen2-VL suite of model, as suggested by the Qwen2 developers, was LLaMA-Factory.¹⁷ Unlike other Large Language Models (LLM) that provide native fine-tuning methods, Qwen2 does not. However, LLaMA-Factory provides a simple process for fine-tuning 100+ different AI models. Most importantly, LLaMA-Factory also provides the ability to perform fine-tuning with multiple images per reference caption.

Before processing our data into the JSON format required by LLaMA-Factory for fine-tuning, we first needed to split our 900 image sets into training and evaluation sets. We required a distinct evaluation for judging how effectively we distilled GPT-4o mini into Qwen2 via fine-tuning. Knowing that we were working with unbalanced classes for our street signs, we made sure to perform balanced class sampling for the two sets. Once separated, we began the process of organizing our images and reference captions into the ShareGPT format required by LLaMA-Factory.¹⁷

The first step to fine-tune Qwen2-VL-7B-Instruct was to process all of the GPT-4o mini captions and image sets into the ShareGPT JSON format required by LLaMA-Factory. To process our captions and images, we employed a pipeline that iterated through the directory of image sets and GPT-4o mini captions to create a conversations and images list. For each sub-directory we added each image to the images list and placed image placeholders along with the original prompt message in one string. The final string would follow the pattern "`< image >< image > ... PROMPT`" and be placed in the "human" portion of the conversations list. The number of `< image >` characters equaled $N+1$, where N is the number of cropped street signs from the main image. Next, the GPT-4o mini caption was added to another string and stored in the "gpt" portion of the conversations list. The conversations and images lists were then combined into one dictionary for each set of images and captions. Thus, the final JSON contained a list of 810 dictionaries, in accordance with the size of the training set.

Once our pipeline finished, we established our training parameters for fine-tuning. For our final fine-tuning run, we opted to use low-rank adaption (LoRA)¹⁸ to help reduce the computation required to fine-tune Qwen. To further improve the efficiency of fine-tuning we made use of a liger kernel.¹⁹ Finally, we used three training epochs with a cosine learning rate of 0.00005. Our final fine-tuning displayed an effective train/loss curve that can be seen in Figure 4 below.

3.4 Final Image Captioning

Once we saved our final fine-tuned Qwen2-VL-7B-Instruct model we could re-caption our 900 image sets to determine how well fine-tuning distilled GPT-4o mini. We were able to re-purpose our original image captioning pipeline to caption our image sets using the Qwen2-VL-7B-Instruct fine-tuned model. Based on our initial spot checks, fine-tuning appeared to vastly improve upon the original Qwen2-VL-7B-Instruct captions. In some cases, fine-tuning even led to an exact replication of the GPT-4o mini caption, only when the image set was included in the training set. Moreover, we observed improved captions for images in the evaluation set as well.

To assess the impact of distilling GPT-4o mini to Qwen2-VL-7B, we had to determine a scoring process that judged the fine-tuned caption on a variety of criteria.

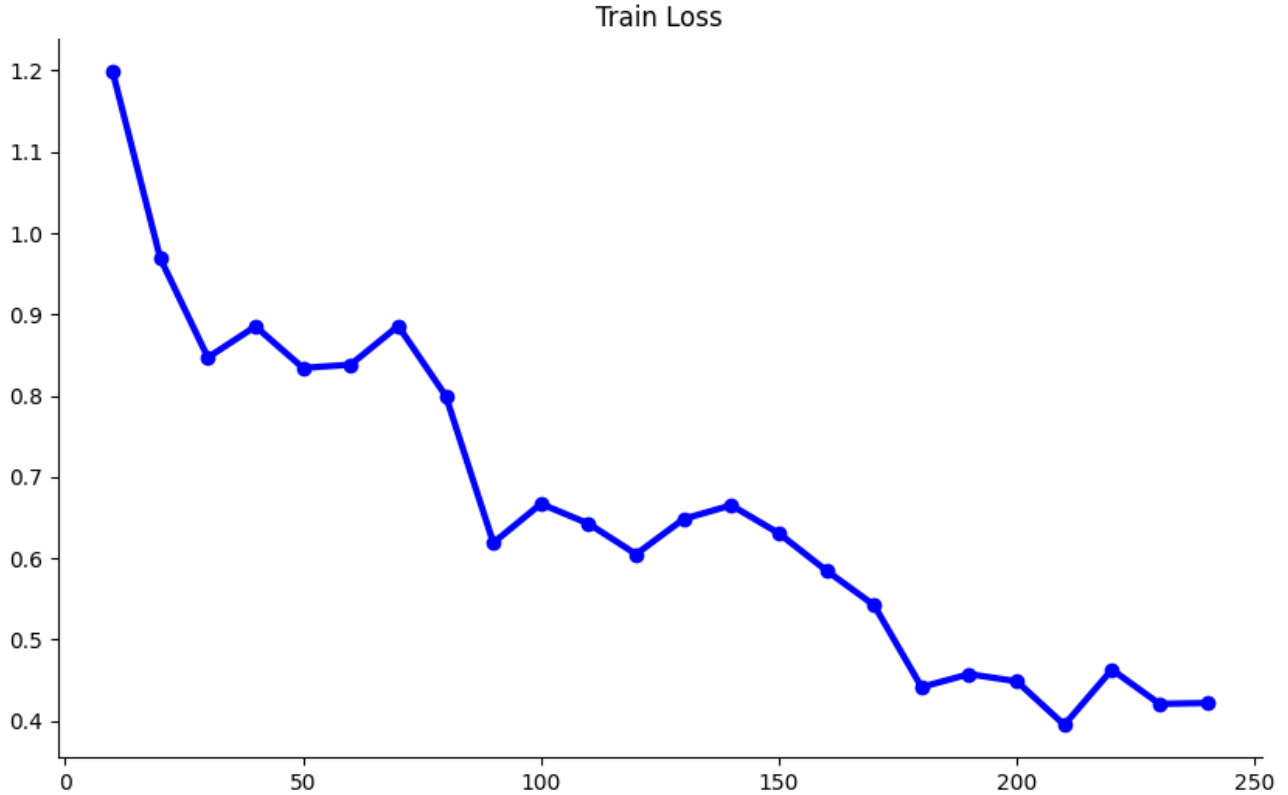


Figure 4. Train/Loss Curve from Weights and Biases.

3.5 LLM-as-a-Judge

Early on in the process of validating the impact of distilling GPT-4o mini we realized the difficulty with assessing the impact of fine-tuning. We needed to simultaneously gauge the similarity in length, meaning, ease of understanding, and other facets of the fine-tuned versus the original GPT-4o mini captions. To aid our scoring process, we determined that using a LLM-as-a-Judge would prove effective, so long as we provided specific prompts for each score’s criteria. In the end, we created five scores that would capture the level of similarity of two captions based on meaning, tone, conciseness, cognitive load, and overall similarity. We chose these five scores because they captured the most important aspects of similarity for our use case of driving instructions based on an image. The most important scores were therefore conciseness and cognitive load to ensure that the driver could just as easily interpret the fine-tuned captions as the original concise GPT-4o mini captions. Meanwhile, the meaning and tone scores determined whether distillation led to an alignment of driving instruction.

To measure these different forms of similarity, we created prompts with the pattern "You will be provided with two image captions. Assign a single score between 1 and 10 for how similar these two captions are. Score: ". This prompt was the one we used to receive a score of overall similarity. Meanwhile, for each other similarity score we simply adjusted the prompt to request a score based on either meaning, tone, conciseness, or ease of understanding (i.e. cognitive load).

For the LLM judge itself, we determined that the best judge would be the Teacher model, GPT-4o mini. To judge each caption pair, we were able to adjust our GPT-4o mini captioning pipeline to instead provide two text captions and the five similarity score prompts. Finally, we averaged the five scores together to get an overall measure of similarity between the GPT-4o mini and fine-tuned Qwen captions. In the following section, we present our findings from scoring all 900 pairs of image captions.

4. RESULTS

4.1 LLM-as-a-Judge Results

The separation of training and evaluation sets provided the necessary basis to determine how effective fine-tuning was in aligning Qwen2 with GPT-4o mini. We hypothesized that balancing the classes between the two sets would provide a sufficient basis in training to result in more situationally aware captions for the evaluation set.

Surpassing our expectations, the evaluation set’s average similarity scores improved by nearly as much as the training set. The average similarity scores improved by 104% and 93% for the training and evaluation sets, respectively. This result indicates that our distillation using the training set was more than sufficient to adjust Qwen’s weights and achieve strong alignment with the capabilities of GPT-4o mini. Figure 5 summarizes this result.

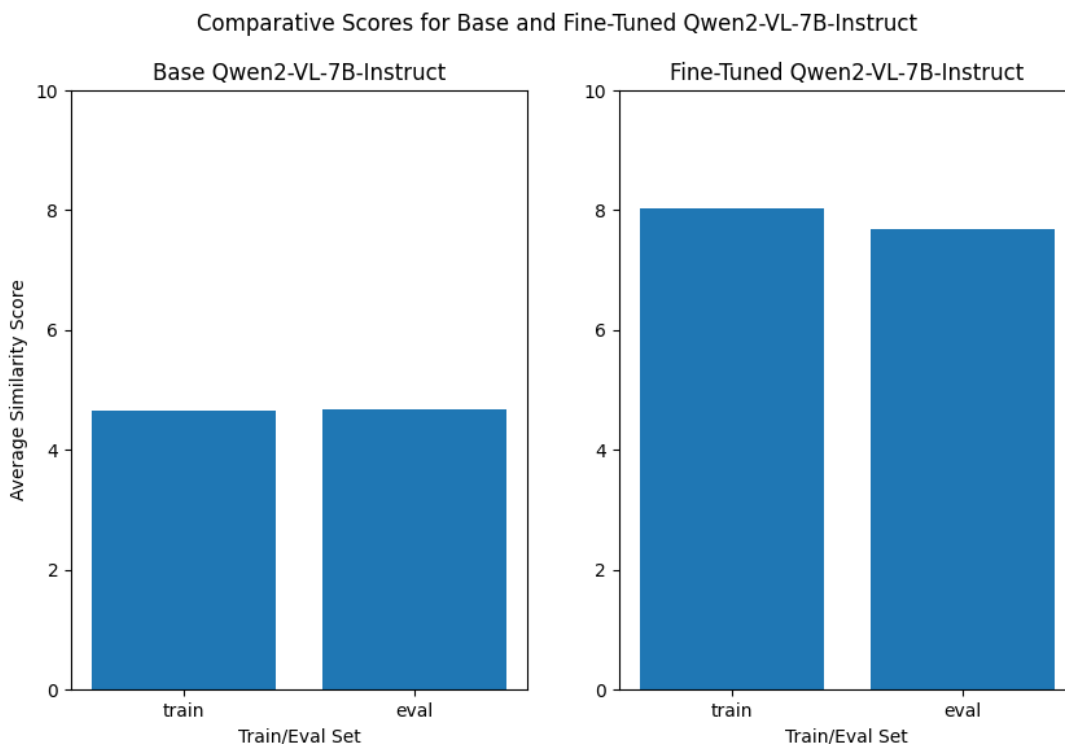


Figure 5. GPT-4o mini acted as the LLM-as-a-Judge to determine the similarity of outputs from both the base Qwen2-VL-7B-Instruct and its fine-tuned variant to the GPT-4o mini outputs across both train and evaluation sets. These findings display the percent improvement in average similarity score from the base model to the fine-tuned one.

During manual spot checks, we found that there was a pattern of incorrect captions for some classes, primarily the prohibitory sign class. Among the other classes, we found that fine-tuning had the greatest impact in improving similarity scores for the danger and "other" street signs for both the training and evaluation sets. The average similarity scores improved by 122% and 107% for the danger class in the train and the evaluation image sets, respectively, while the "other" class scores improved by 137% and 113%. Figure 6 captures these improvements, as well as the remaining class-level average similarity scores from the base and fine-tuned models.

Despite having the highest average similarity score by class, we observed a gap in visual reasoning for a subclass of signs belonging to the prohibitory class. This subclass pertains to street signs that indicate a section of the road with different right of way or no-passing zones. One specific example is shown in Figure 7, which shows a sign that indicates the driver has the right of way at the next intersection. We surmise that these types of signs are misinterpreted by the models because of a lack of understanding for their meaning in their original and fine-tuning training sets. For instance, these signs do not exist in the United States.

Comparative Scores for Base and Fine-Tuned Qwen2-VL-7B-Instruct By Class

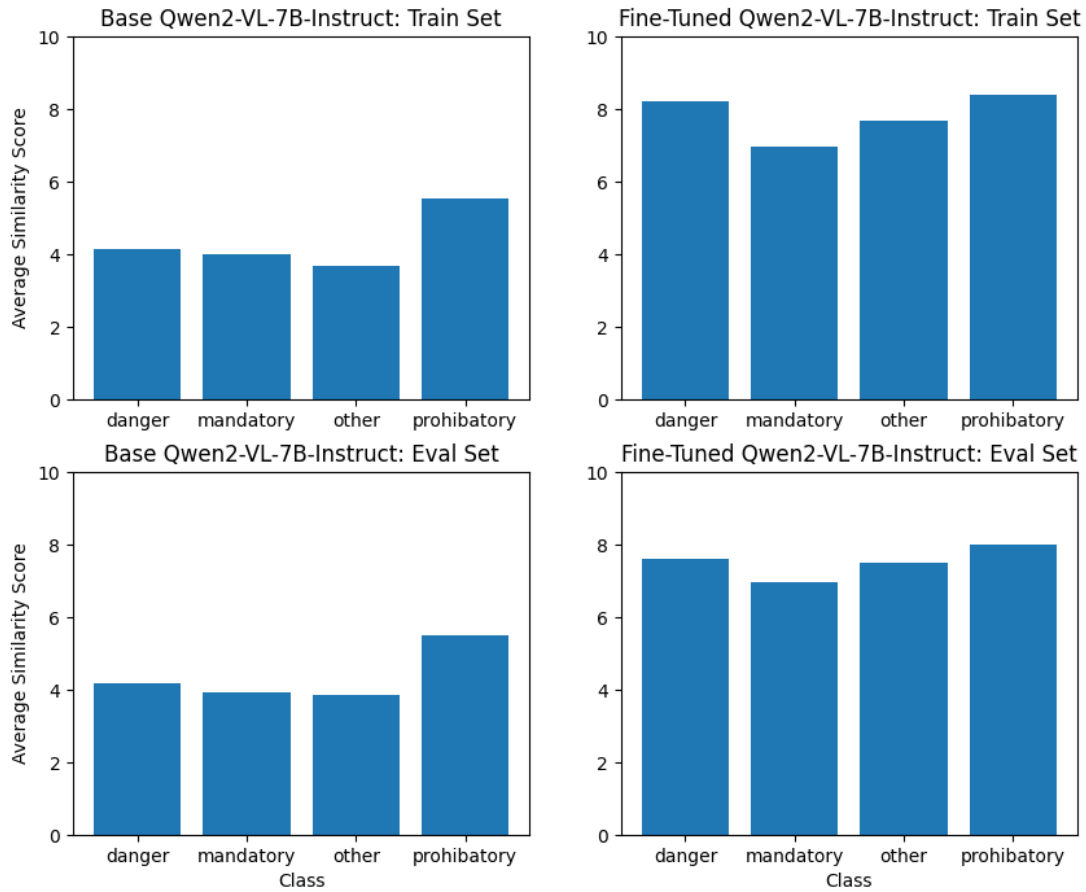


Figure 6. LLM-as-a-Judge average similarity scores are shown here for each class in our street sign dataset, across the training and evaluation sets and both the base and fine-tuned Qwen2-VL-7B-Instruct models.

In Figure 7 above, we observe that the mandatory and "other" classes have the weakest alignment with GPT-4o mini across both train and evaluation sets. Our manual spot checks confirmed that both the Qwen and GPT models struggled to correctly interpret these classes. From these investigations, we found that street signs indicating the need to drive around a median or other obstacle on the roadway have the lowest average scores. In these cases, our models interpret the sign similar to a mandatory turn, rather than a mandatory avoidance. Figure 8 shows an example of such an image from our dataset. In this case, we posit that the model is oversimplifying the meaning of an arrow that is pointing in a direction, even if it is slanted. It is entirely possible that the images with signs indicating a mandatory turn left or right are affecting the model's ability to correctly interpret this slanted version.

4.2 Fine-Tuning Results

As suggested by the scores from the previous section, captions noticeably improved after fine-tuning across both the training and evaluation set. However, we noticed an emergent behavior of the fine-tuned model having more accurate captions than GPT-4o mini in several cases. One clear example of this behavior is for one of the prohibitory street signs in the training set that shows a speed limit of 70 km/h, shown in Figure 9.

The initial caption from the base Qwen2 model was quite weak as it stated "The sign indicates that you are



Figure 7. Image 00000 from the German street signs dataset showing a prohibitory street sign.



Figure 8. Image 00255 from the German street signs dataset showing a mandatory street sign.

approaching a roundabout. You should prepare to merge into the roundabout and follow the traffic flow.” Of course, there is no roundabout ahead. The reference caption was much closer, but made a mistake in suggesting that the driver should “Drive slowly and be attentive, as the speed limit is 30 km/h on this road.” Only the fine-tuned Qwen2 model correctly instructs “Drive carefully and maintain a speed of 70 km/h as you navigate the winding road ahead.” Not only does the fine-tuned model provide the correct speed in a visually obstructed setting, it also adds context for the roadway in its reference to the winding road ahead. Such a result suggests the possibility that fine-tuning updated the weights of Qwen2-VL-7B-Instruct in such a way that its visual reasoning was enhanced beyond GPT-4o mini. However, since GPT-4o mini uses UTF-8 encoding it is possible that the cause of this discrepant performance is the tokenization methods for Qwen2 and GPT-4o mini models.

5. CONCLUSION

This study presents the value of distilling a Large Vision Language Model (LVLM) into a smaller VLM to develop a system that is capable of outputting accurate image captions with a relatively small computational cost. A key result that proves the success of fine-tuning is the 93% improvement in average LLM judged similarity score for images in our evaluation set, while the average similarity scores for the training set more than doubled.

During our investigation of the fine-tuned model’s outputs we additionally noted several emergent behaviors. Most notably, there are some cases where the fine-tuned model provides a correct caption despite the Teacher



Figure 9. Image 00018 from the German street signs dataset with significant sun glare.

model’s caption being incorrect. Such a result points to the possible benefits of combining two powerful VLMs through a similar distillation process.

Benefits of this kind of system include cost savings and data privacy. Typically, the cost of using powerful closed-systems depends on the level of usage and the cost per token decided by the system provider. In comparison, open-source models primarily have an upfront cost to the development of the model, such as the cost of developer time and the store of the model’s weights for ongoing inference. Additionally, data privacy may be a point of consideration for organizations seeking to build AI systems. Most commonly, there is considerable proprietary information that the organization wants to train the model on, but not expose to the closed systems for fear of a data breach. On the other hand, open-source systems can be trained and saved locally to avoid any chance of unwanted data sharing.

One area of opportunity for future development within the distillation system we built is to improve several of the street sign classes that were incorrectly captioned by both the Teacher and Student models. We hypothesize that these outputs can be improved either by providing information across other modalities, such as audio visual data, or by supplying human-corrected captions for these edge-case street sign classes.

One point of emphasis throughout this investigation is that smaller models can be fine-tuned to solve complex challenges without the added overhead cost in energy and currency. In the future we would take a defense-based problem and use this process to create a smaller model to test on the problem. We would also like to experiment with the compute and complexity requirements required for our smaller model to do real-time analytics on an edge-capable device.

ACKNOWLEDGMENTS

This paper is an extension of the work completed during the Spring 2025 Independent Study for Data Science (EN.685.801) at Johns Hopkins University, Whiting School of Engineering, Engineering for Professionals, Data Science Master’s Program. Special thanks to Dr. Tony Johnson from Johns Hopkins University, Whiting School of Engineering, Engineering for Professionals Data Science Program, for their support in this Capstone Project.

REFERENCES

- [1] OpenAI, “Introducing chatgpt,” (2022).
- [2] Bordes, F., Pang, R. Y., Ajay, A., Li, A. C., Bardes, A., Petryk, S., Mañas, O., Lin, Z., Mahmoud, A., Jayaraman, B., Ibrahim, M., Hall, M., Xiong, Y., Lebensold, J., Ross, C., Jayakumar, S., Guo, C., Bouchacourt, D., Al-Tahan, H., Padthe, K., Sharma, V., Xu, H., Tan, X. E., Richards, M., Lavoie, S., Astolfi, P., Hemmat, R. A., Chen, J., Tirumala, K., Assouel, R., Moayeri, M., Talattof, A., Chaudhuri, K., Liu, Z., Chen, X., Garrido, Q., Ullrich, K., Agrawal, A., Saenko, K., Celikyilmaz, A., and Chandra, V., “An introduction to vision-language modeling,” (2024).

- [3] Liu, H., Li, C., Wu, Q., and Lee, Y. J., “Visual instruction tuning,” (2023).
- [4] Wang, W., Lv, Q., Yu, W., Hong, W., Qi, J., Wang, Y., Ji, J., Yang, Z., Zhao, L., Song, X., Xu, J., Xu, B., Li, J., Dong, Y., Ding, M., and Tang, J., “Cogvlm: Visual expert for pretrained language models,” (2024).
- [5] Liu, J., Gan, Z., Wang, L., Yin, S., Liu, Z., Liu, C., and Gao, J., “Instructblip: Towards general-purpose vision-language models with instruction tuning,” *arXiv preprint arXiv:2305.06500* (2023).
- [6] Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., Fan, Y., Dang, K., Du, M., Ren, X., Men, R., Liu, D., Zhou, C., Zhou, J., and Lin, J., “Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution,” *arXiv preprint arXiv:2409.12191* (2024).
- [7] Zhu, D., Chen, J., Shen, X., Li, X., and Elhoseiny, M., “MiniGPT-4: Enhancing vision-language understanding with advanced large language models,” in [*The Twelfth International Conference on Learning Representations*], (2024).
- [8] Li, Y., Yue, X., Xu, Z., Jiang, F., Niu, L., Lin, B. Y., Ramasubramanian, B., and Poovendran, R., “Small models struggle to learn from strong reasoners,” *arXiv preprint arXiv:2502.12143* (2025).
- [9] Stephan, A., Zhu, D., Aßenmacher, M., Shen, X., and Roth, B., “From calculation to adjudication: Examining llm judges on mathematical reasoning tasks,” *arXiv preprint arXiv:2409.04168* (2024).
- [10] Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., and Stoica, I., “Judging llm-as-a-judge with mt-bench and chatbot arena,” *arXiv preprint arXiv:2306.05685* (2023).
- [11] Sichkar, V., “Traffic signs dataset in yolo format,” (2020).
- [12] Redmon, J., Divvala, S. K., Girshick, R. B., and Farhadi, A., “You only look once: Unified, real-time object detection,” *CoRR* **abs/1506.02640** (2015).
- [13] Clark, A. and Contributors, “Pillow (pil fork) documentation,” (2023).
- [14] OpenAI, “Chatgpt o1 model api,” (2024).
- [15] NVIDIA, “Nvidia jetson orin,” (2024).
- [16] Zeff, M., “Openai unveils gpt-4o mini—a smaller, much cheaper multimodal ai model,” (2024).
- [17] Zheng, Y., Zhang, R., Zhang, J., Ye, Y., Luo, Z., Feng, Z., and Ma, Y., “LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models,” *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Association for Computational Linguistics (2024).
- [18] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., and Chen, W., “Lora: Low-rank adaptation of large language models,” *CoRR* **abs/2106.09685** (2021).
- [19] Hsu, P.-L., Dai, Y., Kothapalli, V., Song, Q., Tang, S., Zhu, S., Shimizu, S., Sahni, S., Ning, H., and Chen, Y., “Liger kernel: Efficient triton kernels for llm training,” (2025).