Wolf: Dense Video Captioning with a World Summarization Framework (Appendix)

Anonymous Author(s)

A Contributions

We would like to list Wolf Contributions:

1) Framework and Evaluation Metric. We designed a novel world summarization framework, **Wolf**, for video captioning and introduced an LLM-based metric, **CapScore**, to evaluate the quality of captions. The results show that our method significantly improves CapScore.

2) Datasets and Benchmark. We introduce the Wolf benchmark (leaderboard) and four human-annotated benchmark datasets. These datasets include autonomous driving, general scenes from Pexels, robotics videos, and human-annotated captions, collectively referred to as the **Wolf Dataset**.

3) Intended Uses. We believe Wolf can serve as one of the best practices (auto-labeling tool) for creating and curating paired datasets and benchmarks.

4) Hosting, licensing, and maintenance plan. The code, data, and leaderboard will be open-sourced and maintained. Continuous efforts will be made to refine the Wolf Dataset, Wolf codebase, and CapScore. We hope that Wolf will raise awareness about the quality of video captioning, set a standard for the field, and boost community development.

B Pexel Dataset Categories

We categorize videos from pexel into the following types: Travel & Events, Sports, Education, Pets & Animals, People & Blogs, Nonprofits & Activism, News & Politics, Music, Science & Technology, Comedy, Entertainment, Film & Animation, Gaming, Robotics, How to Styles.

C Qualitative Caption Comparison on Interactive Nuscenes Driving Videos

We show the qualitative results of Wolf in Figure 1. We noticed that although GPT-4V is good at recognizing the scenes, capturing temporal information in a video is not ideal. Gemini-Pro-1.5 can capture video information such as "waiting their turn while others proceed through the intersection when it's clear", but it fails to describe the detailed motions. In comparison to these two state-of-the-art approaches, we observed that Wolf not only captures the motion described in Gemini-Pro-1.5 but also successfully captures "vehicles moving in different directions" and "vehicles accelerating and decelerating as they approach and leave the intersection in response to traffic signals or the flow of other vehicles". We also display the details of Figure 4 of the paper (Wolf example for driving videos that focus on interactive operations) in Figure 2.

D Ablation Study on Token Efficiency

It is well-known that the LLMs finetuned with RLHF favor longer response [1], a phenomenon referred to as verbosity issue. To better assess the efficiency of the captions, we performed additional evaluation using the CapScore judge. Specifically, we separate each caption result into sentences,

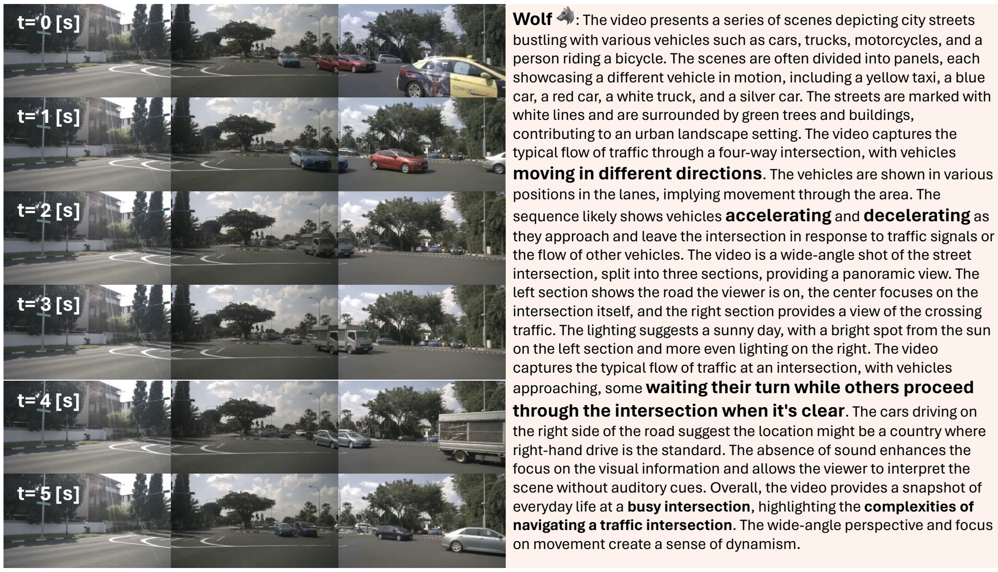

Figure 1: Wolf example for driving that focus on interactive operations. Wolf captions discusses the motion behavior in details and serves as a good reference for autonomous driving. Note: Please refer to the Appendix for our caption comparison with other state-of-the-art methods.

then incrementally use more sentences to form shortened captions, starting from only using the first sentence, to the whole original caption. These shortened captions are scored via CapScore, and we plot the score against the number of tokens used. We show the results in Figure 3.

From the result, we observe that for the better performing models (Wolf, Gemini-Pro-1.5 and GPT-4V) the similarity scores grow with token length when caption lengths are short, but quickly plateau or even drop as the caption lengths get too long. The caption quality score demonstrates quite diverse patterns from different models. GPT-4V maintains a relatively consistent quality score while Gemini-Pro-1.5 and Wolf display better quality when the caption length is short.

E Discussion and Future Works

E.1 Limitations and Optimization.

Wolf is still significantly more cost-effective for autolabeling and captioning than procuring human labels. However, there is an efficiency concern when using an ensemble method. This must be handled with care to ensure that GPU resources are used effectively to mitigate any throughput degradation compared to using single models, even though Wolf offers a significant improvement in caption quality. Modern GPUs are based on a massively parallel pipeline, and our goal is to saturate this pipeline with meaningful work. We consider three primary areas for optimization to make Wolf a unified and efficient framework: Low-Hanging Fruit, Batched Inference, and Model Quantization. For example, we reduce the size of the model weights for model quantization. Recent works [2, 3, 4] have noted that LLMs and VLMs can produce highly accurate results even when their weights are quantized to low bit depths. Therefore, we quantize all constituent models used in Wolf to 4 bits to further improve efficiency. This has two benefits. First, it reduces the bandwidth required for computation. These algorithms work by packing two 4-bit numbers into a single 8-bit type, so when moving data on the GPU, only half the number of bits need to be moved. Since all currently released GPUs support native instructions on 8-bit floating point numbers, the two 4-bit numbers are extracted and expanded by each kernel. In other words, two computations can be performed for every move operation. Next-generation GPUs will natively support 4-bit data types, and we expect further efficiency improvements from having dedicated 4-bit multiply and add instructions. Second, it synergizes with batched inference since the model weights, which are traditionally 16-bit, now only

CogAgent: The picture shows a three-panel scene of a city street. In the first panel, a white truck is driving on the road, and there are several cars and motorcycles on the street. In the second panel, the white truck is still driving on the road, and there are several cars and motorcycles on the street. In the third panel, the white truck is driving on the road, and there are several cars and motorcycles on the street.

VILA-1.5: The video presents a sequence of images showing a street view with a pedestrian crossing. The images feature a white truck parked on the left side of the frame, a pedestrian crossing the street, and a pedestrian crossing sign visible on the right side. The sky is clear with few clouds, and the buildings in the background have a modern architectural style. The street is lined with trees and there are multiple cars parked on the right side of the street. The pedestrian is wearing dark clothing and is seen walking from one side of the street to the other.

GPT-4: It appears that you've provided a series of still images which seem to be sequential frames from a video that captures a scene at a traffic intersection. The visual elements include the following: a road with multiple lanes, white lane markings, and a variety of vehicles, including cars and a truck, navigating the intersection. The vehicles are captured in various positions in the lanes, implying movement through the area. There appears to be a tree-lined median or central island, and trees and buildings can be seen in the background, contributing to an urban landscape setting. The motion behavior depicted across these frames suggests the typical flow of traffic through an intersection. Vehicles are moving in a coordinated manner, adhering to lane directions and traffic signals that are implied rather than visible in the images. The sequence likely shows vehicles accelerating and decelerating as they approach and leave the intersection in response to traffic signals or the flow of other vehicles. The spatial positioning of the vehicles changes progressively, implying a real-time capture of their movements. If the sequence was put together as a continuous video, you would see vehicles entering the frame from one side and exiting on the other, with varying speeds and some vehicles possibly turning, illustrating the dynamic and fluid nature of traffic movement within a regulated urban environment.

Gemini-Pro-1.5: The video is a wide-angle shot of a street intersection. The image is split into three sections, providing a panoramic view. Here's a breakdown: Visual Elements: Perspective: The three-part image creates a wide-angle perspective, simulating the view a driver might have at the intersection. The left section shows the road the (assumed) viewer is on, the center focuses on the intersection itself, and the right section provides a view of the crossing traffic. Color Palette: The video appears to be in color, but the colors are somewhat muted, giving it a realistic but slightly desaturated feel. Lighting: The lighting suggests a sunny day. The left section has a bright spot from the sun, possibly late afternoon, while the right section is more evenly lit. Static Elements: Elements like trees, buildings, lampposts, and road markings provide context and a sense of place. Dynamic Elements: Vehicles, including cars and a truck, move through the intersection, creating a dynamic scene. Narrative Elements/Motion Behavior: Traffic Flow: The video captures the typical flow of traffic at an intersection. Vehicles approach, some wait their turn while others proceed through the intersection when it's clear. Right-Hand Drive: The cars driving on the right side of the road suggest the location might be a country where right-hand drive is the standard. Implied Viewer: While there is no visible camera or person, the perspective of the shot implies the viewer is positioned within a vehicle approaching the intersection. No Sound: The absence of sound enhances the focus on the visual information and allows the viewer to interpret the scene without auditory cues. Overall Impression: The video provides a snapshot of everyday life at a busy intersection. The wide-angle perspective and focus on movement create a sense of dynamism and highlight the complexities of navigating a traffic intersection.

Wolf: The video presents a series of scenes depicting city streets bustling with various vehicles such as cars, trucks, motorcycles, and a person riding a bicycle. The scenes are often divided into panels, each showcasing a different vehicle in motion, including a yellow taxi, a blue car, a red car, a white truck, and a silver car. The streets are marked with white lines and are surrounded by green trees and buildings, contributing to an urban landscape setting. The video captures the typical flow of traffic through a four-way intersection, with vehicles **moving in different directions**. The vehicles are shown in various positions in the lanes, implying movement through the area. The sequence likely shows vehicles **accelerating** and **decelerating** as they approach and leave the intersection in response to traffic signals or the flow of other vehicles. The video is a wide-angle shot of the street intersection, split into three sections, providing a panoramic view. The left section shows the road the viewer is on, the center focuses on the intersection itself, and the right section provides a view of the crossing traffic. The lighting suggests a sunny day, with a bright spot from the sun on the left section and more even lighting on the right. The video captures the typical flow of traffic at an intersection, with vehicles approaching, some **waiting their turn while others proceed through the intersection when it's clear**. The cars driving on the right side of the road suggest the location might be a country where right-hand drive is the standard. The absence of sound enhances the focus on the visual information and allows the viewer to interpret the scene without auditory cues. Overall, the video provides a snapshot of everyday life at a busy intersection, highlighting the complexities of navigating a traffic intersection. The wide-angle perspective and focus on movement create a sense of dynamism.

Figure 2: Comparison of CogAgent, VILA-1.5, GPT-4, Gemini-Pro-1.5, and Wolf on Interactive Nuscenes Driving Videos, Corresponding to Figure 4 of the Paper.

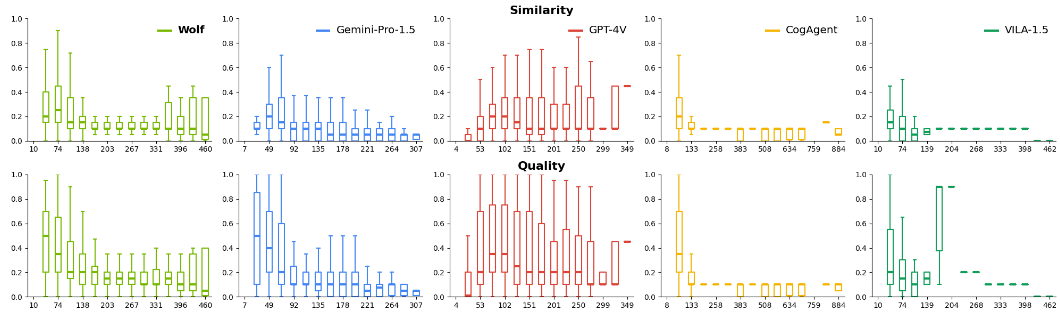

Figure 3: CapScore Caption Similarity and Caption Quality evaluated under varying caption length.

require one quarter of the GPU memory they would ordinarily use. This allows us to fit larger batch sizes on each GPU and process more videos in parallel.

E.2 Wolf Efficiency Optimization

We consider three primary areas: **Low-Hanging Fruit**, **Batched Inference**, and **Model Quantization** as optimizations which make Wolf a unified and efficient framework. Using the optimizations detailed in this section we were able to increase the speed of CogVLM by a factor of approximately 10x (450s/video to 41s/video), VILA throughput was similarly improved to only about 3s per video.

Low-Hanging Fruit. These are primarily systems concerns and work arounds for simplistically written APIs. For example, the off-the-shelf CogVLM [5] and VILA [6] supporting code is heavily based on loading PIL images to present to a huggingface pipeline [7]. In the naive pipeline, videos would need to be decoded and then converted to PIL images before input to the respective pipelines, which in turn convert them to GPU PyTorch [8] tensors. This is extremely inefficient. Instead, we can leverage the hardware video decoder present in modern GPUs to decode the videos directly to GPU tensors and rewrite the preprocessing pipelines to operate on these tensors directly. This has the additional benefit of shifting preprocessing transform work from CPU to GPU.

Batched Inference. Simplifying Wolf into the simplest terms, we are essentially performing repeated neural network inference. Surprisingly, most VLM supporting code is designed to run inference on only a single example at a time. However, just as in other deep-learning problems, there fundamentally no reason why we cannot processes multiple videos at a single time in batches. This step is crucial to maximizing the use of GPU resources. Processing a single example may only use as little as 25% of a modern datacenter GPU which would either increase the time to process a dataset or the number of GPUs required to complete a task in a fixed time budget. We can reimplement more of the supporting code to enable processing batches of as many videos as will fit in GPU memory at a single time yielding a linear speedup in processing. For example, if we can fit batches of 4 in GPU memory we observe a speedup of 4x over processing single examples.

Model Quantization. The final optimization we consider is to reduce the size of the model weights. Several recent works [2, 3, 4] have noted that LLMs and VLMs can produce highly accurate results even when their weights are quantized to low bit-depths. Therefore, we quantize all constituent models used in Wolf to 4-bits to further improve efficiency. This has two benefits. First, it reduces the bandwidth required for computation. These algorithms work by packing two 4-bit numbers into a single 8-bit type, so when moving data on the GPU only half the number of bits need to be moved. Since all currently released GPUs support native instructions on 8-bit floating point numbers, the two 4-bit numbers are extracted and expanded by each kernel. In other words, two computations can be performed for every move operation. Next generation GPUs will natively support 4-bit datatypes and we expect further efficiency improvements from having dedicated 4-bit multiply and add instructions. Next, it synergizes with batched inference since the model weights, which are traditionally 16-bit, now only require one quarter of the GPU memory they would ordinarily use. This allows us to fit larger batch sizes on each GPU and process more videos in parallel.

Model	CLIP-Score ↑	CapScore$_S$↑	CapScore$_Q$↑	N-avg↑
MiniGPT-4	<u>0.601</u>	0.330	0.359	0.19
InstructBLIP	0.599	0.360	0.355	0.18
LLaVA-1.5	<u>0.601</u>	<u>0.385</u>	**0.450**	<u>0.67</u>
mPLUG-Owl2	0.597	**0.397**	0.400	0.49
Qwen2-VL	**0.618**	0.373	<u>0.432</u>	**0.82**

Table 1: Comparison on CapScore and CLIP-Score for text-image alignment. CapScore$_S$ represents CapScore Similarity; CapScore$_Q$ represents CapScore Quality (such as reduced hallucination); N-avg represents noun/verb average. We observe that CapScore aligns with trends observed in other metrics but highlights a larger performance gap between models, suggesting it serves as a more effective evaluation metric. Note: All scores are scaled to the range [0, 1].

Method	Caption Similarity ↑	Caption Quality ↑
CogAgent [5]	0.27	0.30
VILA-1.5-7B [6]	0.35	0.39
Wolf (based on VILA-1.5-7B)	**0.56**	**0.60**

Table 2: Comparison on 4,785 normal Nuscenes videos. The quality of Wolf is consistently better.

E.3 Safety Considerations.

As an ensemble of captioners, Wolf mitigates the possibility of missing out on crucial information in the captions and rectifying any hallucinations that do not agree with the output of most models, which is a fundamental pillar for developing safe autonomous systems, as specified in the functional safety standard ISO 26262 [9]. Beyond the benefits of Wolf, there are still various open questions pertaining to safety of VLM captioners in deployment which we aim to explore more in future: (i) We need to align the captions with the task at hand; e.g., in a driving scenario, a detailed description of the foliage around the road, even if correct, is irrelevant and can potentially act as distractor for the decision maker. (ii) Complementary to the first point, we need to *measure* how well a caption aligns with the task at hand and develop an advanced version of CapScore. (iii) Finally, we need an approach to quantify the confidence we have in the captions by leveraging techniques from learning theory, such as conformal prediction [10]. Most prior work in this direction assumes an MCQ-styled outputs or those where a unique correct answer exists [11, 12], but these approaches do not translate to free-form text descriptions.

F Comparing CapScore with Other Metrics

To verify the efficiency of CapScore, we compare CapScore with the two most widely used captioning scores: 'CLIP-Score' [13] and 'Noun and verb coverage' (N-avg) []. Using CLIP, the *CLIPScore* between the image I and all the generated captions is computed. *Recall@k* is calculated to determine whether the corresponding generated caption y' appears within the top-k most similar captions. N-avg assesses how well a caption y' covers key objects (nouns) and actions (verbs) present in an image by comparing it to the groundtruth y.

Noun coverage is calculated as:

$$\text{Noun Coverage} = \frac{|N(y) \cap N(y')|}{|N(y')|} \tag{1}$$

where $N(y')$ is the set of all nouns in y'. Verb coverage is calculated for verbs likewise.

We evaluate various popular models on a wide-used image dataset COCO Karpathy test set [14]: MiniGPT-4 [15], InstructBLIP [16], LLaVA-1.5 [17], mPLUG-Owl2 [18] and Qwen2-VL [19]. As is shown in Table 1, we observe that CapScore aligns with trends observed in other metrics but highlights a larger performance gap between models, suggesting it serves as a more effective evaluation metric.

G Updated Results and Documentation

We will regularly update Wolf results and documentation on our webpage. Thank you for your time and patience in reviewing our paper!

References

[1] P. Singhal, T. Goyal, J. Xu, and G. Durrett. A long way to go: Investigating length correlations in rlhf. *arXiv preprint arXiv:2310.03716*, 2023.

[2] J. Lin, J. Tang, H. Tang, S. Yang, X. Dang, and S. Han. Awq: Activation-aware weight quantization for llm compression and acceleration. *arXiv preprint arXiv:2306.00978*, 2023.

[3] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.

[4] S. Ma, H. Wang, L. Ma, L. Wang, W. Wang, S. Huang, L. Dong, R. Wang, J. Xue, and F. Wei. The era of 1-bit llms: All large language models are in 1.58 bits. *arXiv preprint arXiv:2402.17764*, 2024.

[5] W. Hong, W. Wang, Q. Lv, J. Xu, W. Yu, J. Ji, Y. Wang, Z. Wang, Y. Dong, M. Ding, and J. Tang. Cogagent: A visual language model for gui agents, 2024.

[6] J. Lin, H. Yin, W. Ping, Y. Lu, P. Molchanov, A. Tao, H. Mao, J. Kautz, M. Shoeybi, and S. Han. Vila: On pre-training for visual language models, 2023.

[7] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[8] J. Ansel, E. Yang, H. He, N. Gimelshein, A. Jain, M. Voznesensky, B. Bao, P. Bell, D. Berard, E. Burovski, G. Chauhan, A. Chourdia, W. Constable, A. Desmaison, Z. DeVito, E. Ellison, W. Feng, J. Gong, M. Gschwind, B. Hirsh, S. Huang, K. Kalambarkar, L. Kirsch, M. Lazos, M. Lezcano, Y. Liang, J. Liang, Y. Lu, C. Luk, B. Maher, Y. Pan, C. Puhrsch, M. Reso, M. Saroufim, M. Y. Siraichi, H. Suk, M. Suo, P. Tillet, E. Wang, X. Wang, W. Wen, S. Zhang, X. Zhao, K. Zhou, R. Zou, A. Mathews, G. Chanan, P. Wu, and S. Chintala. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM, Apr. 2024. doi:10.1145/3620665.3640366. URL https://pytorch.org/assets/pytorch2-2.pdf.

[9] ROHM. ISO 26262: Functional safety standard for modern road vehicles. URL https://fscdn.rohm.com/en/products/databook/white_paper/iso26262_wp-e.pdf.

[10] G. Shafer and V. Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

[11] A. Z. Ren, A. Dixit, A. Bodrova, S. Singh, S. Tu, N. Brown, P. Xu, L. Takayama, F. Xia, J. Varley, et al. Robots that ask for help: Uncertainty alignment for large language model planners. *arXiv preprint arXiv:2307.01928*, 2023.

[12] A. Z. Ren, J. Clark, A. Dixit, M. Itkina, A. Majumdar, and D. Sadigh. Explore until confident: Efficient exploration for embodied question answering. *arXiv preprint arXiv:2403.15941*, 2024.

[13] J. Hessel, A. Holtzman, M. Forbes, R. L. Bras, and Y. Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021.

[14] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

[15] J. Chen, D. Zhu, X. Shen, X. Li, Z. Liu, P. Zhang, R. Krishnamoorthi, V. Chandra, Y. Xiong, and M. Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023.

[16] W. Dai, J. Li, D. Li, A. M. H. Tiong, J. Zhao, W. Wang, B. A. Li, P. Fung, and S. C. H. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *ArXiv*, abs/2305.06500, 2023. URL https://api.semanticscholar.org/CorpusID:258615266.

[17] H. Liu, C. Li, Y. Li, and Y. J. Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024.

[18] Q. Ye, H. Xu, J. Ye, M. Yan, A. Hu, H. Liu, Q. Qian, J. Zhang, and F. Huang. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. In *CVPR*, 2024.

[19] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.