Wolf: Captioning Everything with a World Summarization Framework

Boyi Li[1,2] **Ligeng Zhu**[1,3] **Ran Tian**[1,2] **Shuhan Tan**[1,4] **Yuxiao Chen**[1] **Yao Lu**[1]
Yin Cui[1] **Sushant Veer**[1] **Max Ehrlich**[1] **Jonah Philion**[1,5] **Xinshuo Weng**[1] **Fuzhao Xue**[1]
Andrew Tao[1] **Ming-Yu Liu**[1] **Sanja Fidler**[1,5] **Boris Ivanovic**[1]
Trevor Darrell[2] **Jitendra Malik**[2] **Song Han**[1,3] **Marco Pavone**[1,6]

[1]NVIDIA [2]UC Berkeley [3]MIT [4]UT Austin [5]University of Toronto [6]Stanford University

Abstract

We propose ***Wolf***, a WOrLd summarization Framework for accurate video captioning. Wolf is an automated captioning framework that adopts a mixture-of-experts approach, leveraging complementary strengths of Vision Language Models (VLMs). By utilizing both image and video models, our framework captures different levels of information and summarizes them efficiently. Our approach can be applied to enhance video understanding, auto-labeling, and captioning. To evaluate caption quality, we introduce CapScore, an LLM-based metric to assess the similarity and quality of generated captions compared to the ground truth captions. We further build four human-annotated datasets in three domains: autonomous driving, general scenes, and robotics, to facilitate comprehensive comparisons. We show that Wolf achieves superior captioning performance compared to state-of-the-art approaches from the research community (VILA1.5, CogAgent) and commercial solutions (Gemini-Pro-1.5, GPT-4V). For instance, in comparison with GPT-4V, Wolf improves CapScore both quality-wise by 55.6% and similarity-wise by 77.4% on challenging driving videos. Finally, we establish a benchmark for video captioning and introduce a leaderboard, aiming to accelerate advancements in video understanding, captioning, and data alignment.

1 Introduction

Video captioning is crucial as it facilitates content understanding and retrieval by providing accurate, searchable descriptions. It also provides pairwise data for effective training of foundation models for tasks like video generation, such as Sora [4] and Runaway [35]. However, generating descriptive, accurate, and detailed video captions remains a challenging research problem for several reasons: firstly, high-quality labeled data are scarce. Video captions from the internet can be faulty and misaligned and human annotation is prohibitively expensive for large datasets. Secondly, video captioning is inherently more challenging than image captioning due to the additional complexity of temporal correlation and camera motion. Existing captioning models [14, 45] struggle with temporal reasoning and fail to achieve accurate scene understanding. Thirdly, there is no established benchmark to measure captioning progress. Existing video QA benchmarks [24] are often limited to short answers, making it difficult to measure hallucinations in detailed long captions. Fourthly, the correctness and completeness of the captions are crucial for safety-critical tasks. In the era of LLMs, text descriptions of scenarios used by embodied agents for planning and control become increasingly common [25, 26, 16, 12]. Consequently, a false or incomplete description of the scenario may lead to the decision-making module overlooking a critical object after training on such caption data, resulting

Workshop on Video-Language Models at 38th Conference on Neural Information Processing Systems (NeurIPS 2024).

in safety risks. For instance, missing the presence of a human in the vicinity of a vegetable-chopping manipulator can lead to an injury.

To handle these challenges, we introduce WOrLd summarization Framework (***Wolf***), a novel summarization captioning framework, along with a captioning metric CapScore, and the Wolf captioning benchmark with corresponding datasets. Unlike previous works that utilize a single model to generate captions, we propose to use multiple models to collaborate [15], producing much more accurate captions. By leveraging multiple models, we can provide more fine-grained details while reducing hallucinations. We show that Wolf achieves superior captioning performance compared to state-of-the-art approaches from the research community (VILA [21], CogAgent [14]) and commercial solutions (Gemini-Pro-1.5 [39], GPT-4V [28]). In summary, we have three main contributions:

1. We design the first world summarization framework **Wolf** for video captioning and introduce an LLM-based metric **CapScore** for evaluating the quality of captions. We have further verified that CapScore aligns with human evaluations. The results show that our method improves CapScore by a large margin.
2. We introduce Wolf benchmark and four human-annotated benchmark datasets. These datasets include autonomous driving, general scenes from Pexels, and robotics videos, along with human-annotated captions, referred to as the **Wolf Dataset**.
3. The code, data and leaderboard will be open-sourced and maintained on the our webpage. Continuous efforts and improvements will be made to refine the Wolf Dataset, 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.

2 Related Works

Image Captioning. Visual language models (VLMs) have shown rapid advancements, achieving leading performance in image captioning tasks, largely due to the success of large language models. CLIP [31] pioneered this field by training a shared feature space for vision and language modalities on image-caption pairs. Building on CLIP, BLIP [18] and BLIP-2 [17] improved performance by aligning the pre-trained encoder with large language models. Following the direction, LLaVA [22] and InstructBLIP [10] demonstrated that jointly training on diverse datasets as an instruction-following task leads to strong generalization across various tasks. VILA [21] highlighted the importance of pre-training with diverse data, and therefore significantly scaled up the pre-training dataset. Kosmos-2 [30] and PaLI-X [7] further introduced pseudo-labeling bounding boxes from open-vocabulary object detectors to scale up the size of pre-training dataset.

Video Captioning. As image-based VLMs are not specifically trained with video data, they are limited in describing details present in the video data. To improve video captioning, PLLaVa [43] builds on top of LLaVa and introduced a parameter-free pooling strategy to enhance the caption quality. Video-llava [19] achieves state-of-the-art performance on several benchmarks by conducting joint training on images and videos, thereby learning a unified visual representation. Additionally, Video-LLama [45] incorporates both video and audio into LLMs by introducing two Q-formers to extract features. Vid2seq [44] conducts large-scale pre-training with narrated videos for dense video captioning. Meanwhile, MV-GPT [36] employs an automated speech recognition (ASR) model to provide additional labeling for the videos.

LLM-based Summarization. Recently many works have found that it is efficient to summarize useful information using LLMs. For example, LLaDA [16] can provide users with helpful instructions based on the user request and corresponding traffic rules in the desired location. OpenAI team finds re-captioning [3] via LLMs can be very helpful.

3 Wolf: Captioning Everything with a World Summarization Framework

We propose Wolf, which is an automated captioning summarization framework that adopts a mixture of experts approach to generate long, accurate, and detailed captions for videos. Figure 1 provides an overview of our framework. In this paper, we use CogAgent [14], GPT-4V [25] to generating image-level captions, and use VILA-1.5 [21], Gemini-Pro-1.5 [39] to generate video captions.

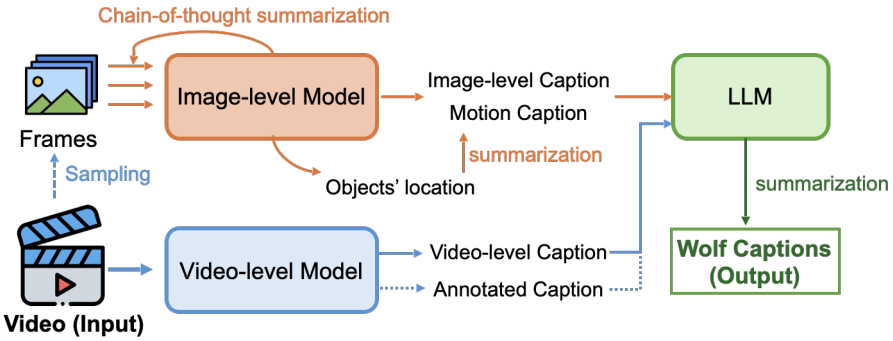


Figure 1: Overview of proposed Wolf framework. Wolf utilizes both image-level and video-level models to generate diverse and detailed captions, which are then summarized for cross-checking.

Chain-of-thought Summarization in Image-level Models. As image-level models (image-based VLMs) have been pre-trained with a larger amount of data than video-level models (video-based VLMs), we first use image-based VLMs to generate captions. We design a Chain-of-thought program to obtain video captions from image-level models. As illustrated in Figure 1, we first split the video into sequential images, sampling two key-frames every second. We start by feeding Image 1 into the Image-level Model to obtain Caption 1, where we require the model to generate detailed scene-level information and object locations. Given the temporal correlation between key frames in a video, we then feed both Caption 1 and Image 2 into the model to generate Caption 2. By repeating this procedure, we generate captions for all sampled frames. Finally, we use GPT-4 to summarize the information from all captions with the prompt "*Summarize all the captions to describe the video with accurate temporal information*".

LLM-based Video Summarization. Besides obtaining the captions from image-level models, we then summarize all captions into one. We use the prompt "*Please summarize on the visual and narrative elements of the video in detail from descriptions from Image Models (Image-level Caption and Motion Caption) and descriptions from Video Models (Video-level Caption)*". Optionally, we can also add the annotated caption to the summarization. Based on this simple scheme, Wolf can capture a rich variety of details of the video and reduce hallucinations (in Figure 2). We assume this is because the model can compare the captions and reduce redundant and hallucinated information. After obtaining the descriptions from the image-level and video-level models, we next apply the prompt "*Please describe the visual and narrative elements of the video in detail, particularly the motion behavior*".

4 Wolf Benchmark: Benchmarking Video Captioning

To showcase the effectiveness of Wolf, we constructed four distinct datasets. These include two autonomous driving video captioning datasets based on the open-sourced NuScenes [6] dataset (Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International Public License), a general daily video captioning dataset from Pexels [1], and a robot manipulation video captioning dataset from an open-source robot learning dataset [29]. These benchmark datasets are tailored to assess the caption model's scene comprehension and its behavior understanding capabilities, both of which are vital for auto-labeling in embodied AI tasks. All captions were generated using a combination of ground truth information, rule-based heuristics, human labeling, and GPT-based rewriting.

4.1 Benchmark Dataset Curation

4.1.1 Autonomous Driving Dataset

High-quality captions of driving videos are crucial not only for training video generation models but also for training VLMs to interpret the dynamic traffic environment. The NuScenes dataset is a large-scale collection of driving videos designed to accelerate autonomous driving research. It features 1,000 annotated scenes from Boston and Singapore. Each scene consists of a 20-second

[1] https://www.pexels.com/

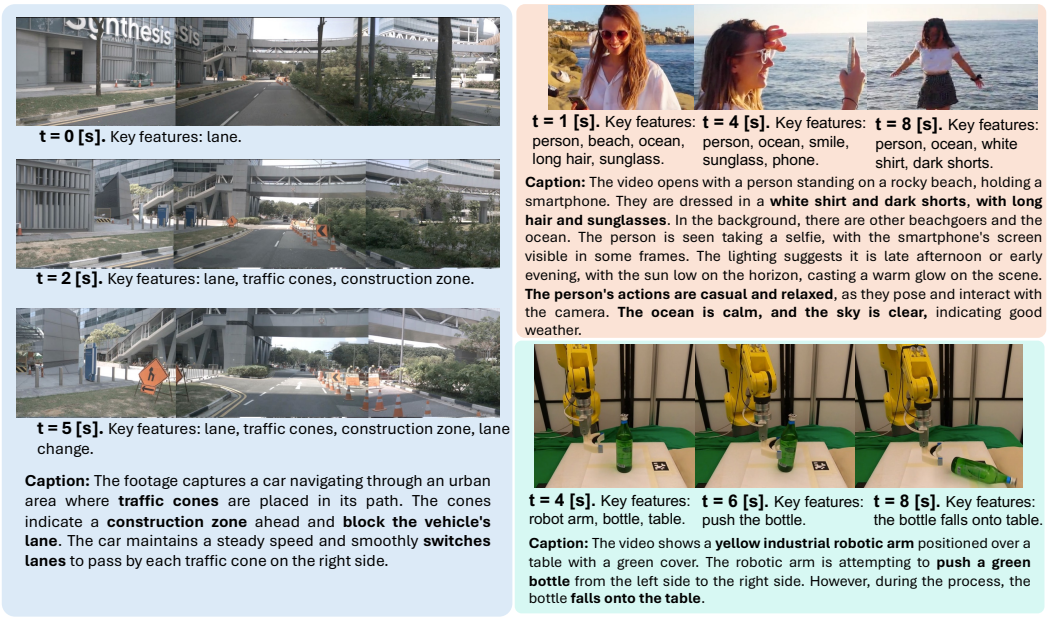


Figure 2: Wolf Dataset examples. We display the videos and corresponding human-annotated captions of autonomous driving (*Left*), Pexels (*Top-Right*), and Robot learning video dataset (*Bottom-Right*), totaling 25.7 hours for now, and the dataset size will be regularly updated and expanded.

driving video clip that provides an ego-centric view from the ego vehicle. We split each scene into 5-second segments and provide the corresponding captions. Our captions emphasize the high-level driving behavior of the ego vehicle to stress test the scene understanding ability and the behavior understanding ability of the captioning model. Our dataset contains **500 intensely interactive video-caption pairs** (≈0.7 hours) in which the ego vehicle is involved in intense interactions with its surrounding traffic agents (such as navigating around construction zones and overtaking static obstacles) and **4785 normal driving scene video-caption pairs** (≈6 hours). Our caption generation process consists of three steps: i) agent-level motion annotation, ii) ego-centric interaction annotation, and iii) GPT-rewriting.

Agent-level motion annotation. The NuScenes dataset provides full annotation of the traffic elements in each scene, including the 3-D bounding box and categories of traffic elements, and semantic map information. Similar to [40], we leverage this ground-truth information and the lane-topology information [27] to annotate both the speed and angular motion characteristics of the ego vehicle and other traffic participants within a video clip. Specifically, we categorize agent actions into 11 types such as Stopping, Accelerating, Decelerating, Lane Changes, Turns, and more, based on their observed movements and behaviors.

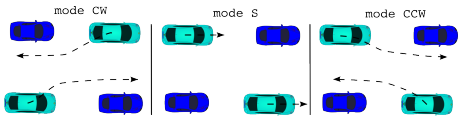


Figure 3: Illustration of homotopy types of different relative motions between a pair of vehicles.

Ego-centric interaction annotation. We are also interested in the ego vehicle's interaction with the other traffic participants (e.g., crossing pedestrians, blocking traffic cones, etc.) shown in the video clip. To efficiently caption the interaction, we leverage two types of categorical modes to describe the lane-relationship between a traffic participant and the ego vehicle *(agent-ego lane mode)* and the relative motion between a traffic participant and the ego vehicle *(homotopy)*[8]. Agent-ego lane mode at a time step t encodes the topology relationship between the ego's current lane and the traffic agent's lane, including: *LEFT*, *RIGHT*, *AHEAD*, *BEHIND*, and *NOTON*, where *NOTON* describes that the traffic agent is not on any derivable lanes in the scene (e.g., a parked vehicle in a parking lot). To compute the agent-ego lane mode for each traffic agent, we follow [8] to first identify the lane on which each agent is located and then leverage the lane topology map to annotate the agent-ego lane mode. We project the agent's center to the lane polyline and use its relative position in the local Frenet frame to determine its lane association. Homotopies describe the relative motion between a pair of agents shown in the video, including: [*S*, *CW*, *CCW*] (*static, clockwise, counterclockwise*), as shown in Figure 3.

GPT-rewriting. Combining agent-ego lane mode, homotopy, agent ground truth state information, and scene context information (e.g., ego is located near intersection) together, we can leverage heuristics to annotate the interaction shown in the video clip. For example, in a video clip, a static object's agent-ego lane mode changes from *AHEAD*, to *LEFT*, to *BEHIND*, and the ego vehicle's first performs *RIGHT-LANE-CHANGE*, *KEEP-LANE*, then *LEFT-LANE-CHANGE*, indicating the ego vehicle overtakes that object from the ego vehicle's left side. We identified 6 interaction categories from the NuScenes dataset: 1) bypass blocking traffic cones to navigate around construction zone; 2) yield to crossing pedestrians; 3) yield to incoming vehicles; 4) overtake traffic agents via straddling the lane dividers; 5) overtake traffic agent via lane-change; 6) other non-intensive interactions. With both agent-level motion annotation and ego-centric interaction annotation, we use GPT 3.5 to summarize each clip to build the final caption.

4.1.2 Robot Manipulation Dataset

In addition to the driving environment, we collect **100 robot manipulation videos** (each has a length ranging from 5 seconds to 1 minute) from [29] that demonstrate complex robot manipulations (e.g., pick and place, push, ect.) in various environments, including kitchen, office, lab, and open world. We manually caption each video. The captions focus on the description of the scene and the interaction between the robot and the objects (see the example in Figure 2).

4.1.3 Pexels Dataset

To evaluate caption models in general daily environments, we further collect high quality (360p to 1080p) videos from Pexels [2]. It consists of **473 high-quality videos** sourced globally, where each video has a length varying between 10 seconds and 2 minutes and the content includes 15 popular categories (details in Appendix A.2). This diversity not only adds depth to our dataset but also provides a wide range of scenarios and contexts for our analysis.

4.2 Evaluation Metric and Leaderboard

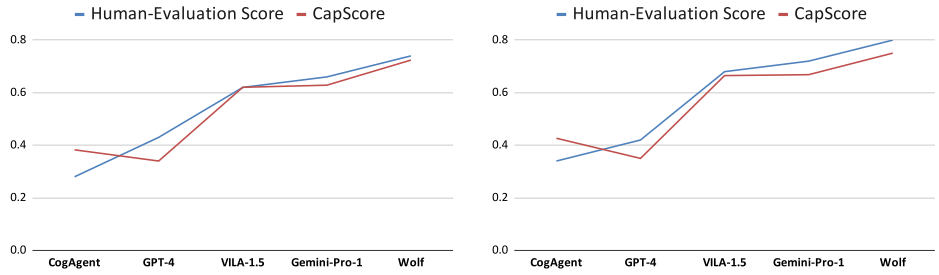
4.2.1 CapScore: Evaluating Captions with LLMs

Video captioning has been an ill-posed problem since there is no metric to evaluate the quality of captions and the alignment between the video and the caption. Inspired by BERTScore [46] and CLIPScore [13], we introduce **CapScore** (Captioning Score), a quantitative metric to use LLMs to evaluate the similarity between predicted and human-annotated (ground-truth) captions. We tried both GPT-4 and LLama 3.1 as our LLM to summarize the captions. We noticed that GPT-4 can always obtain stable results over 3 runs. However, for LLama 3.1, the results varied over different runs. We tried to lower the temperature (from 0.9 to 0.5) to make the inference stable, however, we noticed that the scores are not consistent with human evaluation. Therefore we select GPT-4 as our LLM to conduct the experiments. Assume we have 6 captions, we feed all the captions into GPT-4 and add the prompt "*Can you give a score (two decimal places) from 0 to 1 for captions 1, 2, 3, 4 and 5, indicating which one is closer to the ground truth caption (metric 1) and which contains fewer hallucinations and less misalignment (metric 2)? Please output only the scores of each metric separated only by a semicolon. For each metric, please output only the scores of captions 1, 2, 3, 4 and 5 separated by commas, in order—no text in the output.* ". We ask GPT-4 to output two scores: caption similarity and caption quality.

Caption Similarity. Caption similarity is based on how well each caption aligns with the ground truth description on a scale from 0 to 1, considering the key criteria mentioned. GPT-4 lists the requirements that affect the score: this metric measures how similar each caption is to the ground truth caption. The evaluation focuses on the content and context described in the captions, assessing whether they capture the main themes and details of the ground truth.

Caption Quality. Caption quality evaluates whether the caption contains reduced hallucination and mistakes compared to the ground truth captions on a scale from 0 to 1. GPT-4 lists the criteria that affect the score: this metric evaluates the accuracy and relevance of each caption, identifying any extraneous or incorrect details (hallucinations). Captions with fewer hallucinations and better alignment receive higher scores.

[2] https://www.pexels.com/



(a) Comparison on Caption Similarity. (b) Comparison on Caption Quality.

Figure 4: Comparisons on Human-Evaluation Score and CapScore.

4.2.2 Human-Evaluation Score and CapScore

Through our experiments, we find that GPT-4 is very robust for calculating the scores. We have run the experiments for 1-3 times, the results appear to be stable and less than 0.05 changes. To alleviate concerns related to human alignment and correlation, we randomly selected 10 users to evaluate our set of 100 robotics videos, as detailed in Table 1 of the paper. The evaluators were presented with the videos, the generated captions, and the corresponding ground truth captions. We asked them to assign human-evaluation scores based on the CapScore standard, with the following prompt: "*After reviewing the video and all the captions, please assign the caption similarity and caption quality score (floating point values) from 0 to 1 for different captions, indicating which caption is closest to the ground truth (caption similarity) and which one has fewer hallucinations and less misalignment (caption quality).*" The visual results are shown in Table 1, and we also show the corresponding visual comparison in Figure 4.

Method	Caption Similarity ↑		Caption Quality (eg. reduced hallucination) ↑	
	Human-Evaluation Score	CapScore	Human-Evaluation Score	CapScore
CogAgent [14]	0.28	0.38	0.34	0.43
GPT-4V [1]	0.43	0.34	0.42	0.35
VILA-1.5 [21]	0.62	0.62	0.68	0.67
Gemini-Pro-1.5 [39]	0.66	0.63	0.72	0.67
Wolf	**0.74**	**0.72**	**0.80**	**0.75**

Table 1: Comparison of Human-Evaluation Score and CapScore on 100 Wolf Robotics Videos.

4.2.3 Benchmarking Video Captioning

As far as we know, no standard evaluation benchmarks have been established for video understanding and captioning. To accelerate the advancement of this field, we have developed the first leaderboard for video captioning. As LLM evaluation has become increasingly popular [9], we realized the lack of a standard platform to evaluate VLM's performance on video understanding. We assume this is due to the difficulty of collecting ground-truth captions that accurately align with videos. We will make our leaderboard accessible to everyone.

5 Experiments

5.1 Experimental Setup

Data Setup. We use four sets of data to evaluate the validity of Wolf: 1) 500 Nuscences Interactive Videos; 2) 4,785 Nuscences Normal Videos; 3) 473 general videos and 4) 100 robotics videos. We extract 2 frames per second for autonomous driving videos. For robotics videos, we extract 1 frame per second. For short videos that sample less frames, we will increase `fps` to capture more details.

Comparison Setup. We use our proposed CapScore to evaluate the similarity between predicted and ground truth captions. CogAgent and GPT-4V are image-level methods, so we upload sequential frames into the model to obtain the output. VILA-1.5 and Gemini-Pro 1.5 are video-based methods, so we directly feed a video into the model. As for the prompt for each captioning model, we

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 5: Wolf example for driving videos that focus on interactive operations. Wolf captions discusses the motion behavior in details and serves as a good reference for autonomous driving. Additionally, we provide a webpage where you can view caption comparisons generated by Wolf here: link.

use "*elaborate on the visual and narrative elements of the video in detail, particularly the motion behavior*". We compare with four state-of-the-art image-level and video-level captioning method CogAgent [14], GPT-4V [1], VILA-1.5 [21] and Gemini-Pro-1.5 [39]. As for CogAgent, we feed the middle frame of the video into the model to obtain the captions. As for GPT-4V, we uniformly sample 16 frames from a video and feed the sequential images into the model to obtain captions. As for VILA-1.5 and Gemini-Pro-1.5, we feed the video into the model to obtain the captions.

5.2 Qualitative Results

To illustrate enhanced captioning ability by Wolf, we show the qualitative results in Figure 5 (please check details in Appendix). 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 captions the motion behavior of "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".

5.3 Quantitative Results

We compare Wolf with various state-of-the-art captioning models and display the results on 4 datasets in Table 2 and 3. In the default setting, Wolf uses CogAgent, GPT-4V, VILA-1.5, and Gemini-Pro-1.5 as Video-level models. Due to the running cost, we use Wolf (based on VILA-1.5) on the Nuscenes Normal dataset, which only uses CogAgent and VILA-1.5. We notice that existing image-level models fail to capture the temporal information in detail. Video-level models perform better, while Wolf can achieve the best results compared to all state-of-the-art captioning models.

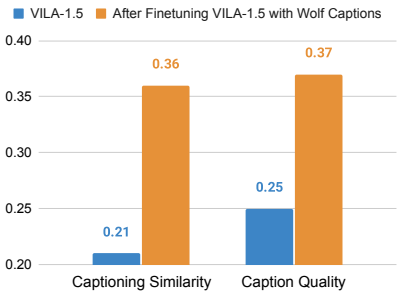


Figure 6: Comparison between VILA-1.5 and fine-tuned VILA-1.5 with Wolf provided captions. on 500 highly interactive Nuscenes videos.

Method	Caption Similarity ↑			Caption Quality (eg. reduced hallucination) ↑		
	Nuscenes	Pexels	Robotics	Nuscenes	Pexels	Robotics
CogAgent [14]	0.18	0.68	0.38	0.24	0.72	0.43
GPT-4V [1]	0.31	0.72	0.34	0.36	0.75	0.35
VILA-1.5 [21]	0.21	0.85	0.62	0.25	0.86	0.67
Gemini-Pro-1.5 [39]	0.42	0.87	0.63	0.45	0.87	0.67
Wolf	**0.55**	**0.88**	**0.72**	**0.56**	**0.89**	**0.75**

Table 2: Comparison on 500 highly interactive (difficulty and challenging) Nuscenes videos, 473 Pexels videos and 100 robotics videos. The best and second results are highlighted with **bold** and underline. Our Wolf exhibits better performance than both open- and closed-source models.

Method	Caption Similarity ↑	Caption Quality (eg. reduced hallucination) ↑
CogAgent [14]	0.27	0.30
VILA-1.5 [21]	0.35	0.39
Wolf (based on VILA-1.5)	**0.56**	**0.60**

Table 3: Comparison on 4,785 normal Nuscenes videos. The quality of Wolf is consistently better.

5.4 Finetuning Video Captioning Models

To further verify the effectiveness of Wolf, we finetune VILA-1.5 based on Wolf's captions on 4,785 normal Nuscenes videos and evaluate it on 500 highly interactive Nuscenes videos, which have much more difficult captions and complex scenarios. We follow the original VILA's training setup and launch supervised-finetuning with Wolf generated video-caption pairs for one epoch. The training is performed on 8xA100 GPUs with batch size 8. We set the learning rate to 10^{-4} with warmup strategy. No weight decay is applied.

We demonstrate the results in Figure 6, corresponding to Table 2. We observe that finetuning with Wolf boosts the model performance to 71.4% on caption similarity and 48.0% on caption quality, which outperforms GPT-4V and approaches Gemini-Pro-1.5. This suggests that Wolf captions can be easily applied to push VLMs' performance to a higher level.

5.5 Ablation Study on Video-level Model Selection

To further evaluate how various video-level models affect the performance, we conduct an ablation study on the components of the models in Table 4. We first compare the caption from the middle frame of CogAgent with Wolf CogAgent Caption based on the chain-of-thought approach. The chain-of-thought procedure could largely improve the video understanding quality from an image-level model such as CogAgent. Then we compare Wolf with various combinations of video captions. We notice that Wolf consistensly shows better CapScore as it incorporates additional video captions.

5.6 Comparison of Finetuned Models

While it is difficult to directly and scalable measure the quality of captions, we compare the same model (VILA-1.5-13B) trained w/ Wolf captions and w/o Wolf captions to study the effectiveness. We benchmark the WOLF-finetuned models on ActivitNet [5] and MSRVTT [42] (suggested by reviewer #p7DT) and display the results in Table 5.

Method	Caption Similarity ↑	Caption Quality (eg. reduced hallucination) ↑
CogAgent	0.18	0.24
Wolf CogAgent part (chain-of-thought)	**0.26**	**0.32**
Wolf (based on VILA-1.5)	0.35	0.37
Wolf (based on VILA-1.5+Gemini-Pro-1.5)	0.48	0.49
Wolf (based on VILA-1.5+Gemini-Pro-1.5+GPT-4V)	**0.55**	**0.56**

Table 4: Ablation study on 500 highly interactive Nuscenes videos.

	ActivityNet	MSRVTT
VILA-1.5-13B	54.7	60.2
VILA-1.5-13B (fine-tuned with Wolf)	**55.2**	**60.9**

Table 5: QA Accuracy Comparison of fine-tuned model on Activity and MSRVTT datasets.

6 Discussion and Future Works

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 like ours. This must be handled with great 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. Several recent works [20, 11, 23] 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 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. Please check our appendix for more explanations.

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 [34]. 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 an approach 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 a method 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 [37]. Most prior work in this direction assumes an MCQ-styled outputs or those where a unique correct answer exists [33, 32], but these approaches do not translate to free-form text descriptions.

7 Conclusion

In this work, we propose Wolf, a captioning framework to annotate any video automatically with significant data alignment improvement. We find out that adopting a mixture of captioning models and summarization can largely boost the quality of the captions. This enables obtaining long, detailed, and accurate video captioning. Since Wolf enables understanding videos from various aspects, it can be efficiently used to understand challenging videos such as multiview driving videos. Beyond that, we set up a leaderboard to boost the development of video captioning, which preserves a guarantee for data alignment. We will also set up a thorough library that contains different types of videos with high-quality captions, regional information such as 2D or 3D bounding boxes and depth, and multiple object motions.

Acknowledgements

We thank Jim Fan, Yuke Zhu, and Jan Kautz for their insightful feedback and the helpful discussions.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, CK Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Michael Suo, Phil Tillet, Eikan Wang, Xiaodong Wang, William Wen, Shunting Zhang, Xu Zhao, Keren Zhou, Richard Zou, Ajit Mathews, Gregory Chanan, Peng Wu, and Soumith 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, April 2024.

[3] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023.

[4] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024.

[5] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015.

[6] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

[7] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. Pali-x: On scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*, 2023.

[8] Yuxiao Chen, Sander Tonkens, and Marco Pavone. Categorical traffic transformer: Interpretable and diverse behavior prediction with tokenized latent. *arXiv preprint arXiv:2311.18307*, 2023.

[9] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, et al. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*, 2024.

[10] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Albert Li, Pascale Fung, and Steven C. H. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *ArXiv*, abs/2305.06500, 2023.

[11] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.

[12] Yan Ding, Xiaohan Zhang, Chris Paxton, and Shiqi Zhang. Task and motion planning with large language models for object rearrangement. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2086–2092. IEEE, 2023.

[13] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021.

[14] Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, and Jie Tang. Cogagent: A visual language model for gui agents, 2024.

[15] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[16] Boyi Li, Yue Wang, Jiageng Mao, Boris Ivanovic, Sushant Veer, Karen Leung, and Marco Pavone. Driving everywhere with large language model policy adaptation. 2024.

[17] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[18] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022.

[19] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection, 2023.

[20] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, and Song Han. Awq: Activation-aware weight quantization for llm compression and acceleration. *arXiv preprint arXiv:2306.00978*, 2023.

[21] Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models, 2023.

[22] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. 2023.

[23] Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit llms: All large language models are in 1.58 bits. *arXiv preprint arXiv:2402.17764*, 2024.

[24] Muhammad Maaz, Hanoona Abdul Rasheed, Salman H. Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv*, abs/2306.05424, 2023.

[25] Jiageng Mao, Yuxi Qian, Hang Zhao, and Yue Wang. Gpt-driver: Learning to drive with gpt. *arXiv preprint arXiv:2310.01415*, 2023.

[26] Jiageng Mao, Junjie Ye, Yuxi Qian, Marco Pavone, and Yue Wang. A language agent for autonomous driving. *arXiv preprint arXiv:2311.10813*, 2023.

[27] Alexander Naumann, Felix Hertlein, Daniel Grimm, Maximilian Zipfl, Steffen Thoma, Achim Rettinger, Lavdim Halilaj, Juergen Luettin, Stefan Schmid, and Holger Caesar. Lanelet2 for nuscenes: Enabling spatial semantic relationships and diverse map-based anchor paths. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 3247–3256, June 2023.

[28] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[29] Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

[30] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023.

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *arXiv: Computer Vision and Pattern Recognition*, 2021.

[32] Allen Z Ren, Jaden Clark, Anushri Dixit, Masha Itkina, Anirudha Majumdar, and Dorsa Sadigh. Explore until confident: Efficient exploration for embodied question answering. *arXiv preprint arXiv:2403.15941*, 2024.

[33] Allen Z Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, et al. Robots that ask for help: Uncertainty alignment for large language model planners. *arXiv preprint arXiv:2307.01928*, 2023.

[34] ROHM. ISO 26262: Functional safety standard for modern road vehicles.

[35] Runway. Gen-3 alpha. https://runwayml.com/ai-tools/gen-3-alpha/, 2024. Accessed on [Insert Date].

[36] Paul Hongsuck Seo, Arsha Nagrani, Anurag Arnab, and Cordelia Schmid. End-to-end generative pretraining for multimodal video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17959–17968, 2022.

[37] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

[38] Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. A long way to go: Investigating length correlations in rlhf. *arXiv preprint arXiv:2310.03716*, 2023.

[39] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[40] Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Chenxu Hu, Yang Wang, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models. *arXiv preprint arXiv:2402.12289*, 2024.

[41] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[42] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016.

[43] Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. Pllava: Parameter-free llava extension from images to videos for video dense captioning. *arXiv preprint arXiv:2404.16994*, 2024.

[44] Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10714–10726, 2023.

[45] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023.

[46] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2019.

A Appendix

A.1 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 on our webpage. 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.

A.2 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.

A.3 Reproducibility Checklist

Datasets Curation. We have provided dataset curation details in "Sec 4.1 Wolf Dataset Curation" of the main paper. The main contribution of the Wolf dataset is its human-annotated captions. Wolf is built upon existing data from Nuscenes, videos from the Pexels website (publicly available, downloaded, and used for free), and robot manipulation videos collected from the Open X-Embodiment dataset [29] (under a CC License). Since all the videos have already been or will be released publicly, we can directly follow the instructions to download and use them.

Evaluation Procedures. We have provided evaluation procedures in "Sec 4.2.1 CapScore: Evaluating Captions with LLMs" of the main paper. Since CapScore compares captions from different methods simultaneously, it can guarantee the quality of the evaluation. In our paper, we ran all the experiments three times and took the average score in the reported tables and figures of the paper.

A.4 Qualitative Comparison on Interactive Nuscenes Driving Videos

We display the details of Figure 4 of the paper (Wolf example for driving videos that focus on interactive operations) in Figure 7. The corresponding video (5cca256ed9ac40a1ac2f91fe485a4d5c.mp4) can be viewed here.

A.5 Ablation Study on Token Efficiency

It is well-known that the LLMs finetuned with RLHF favor longer response [38], 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, 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 8.

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.

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 7: 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.

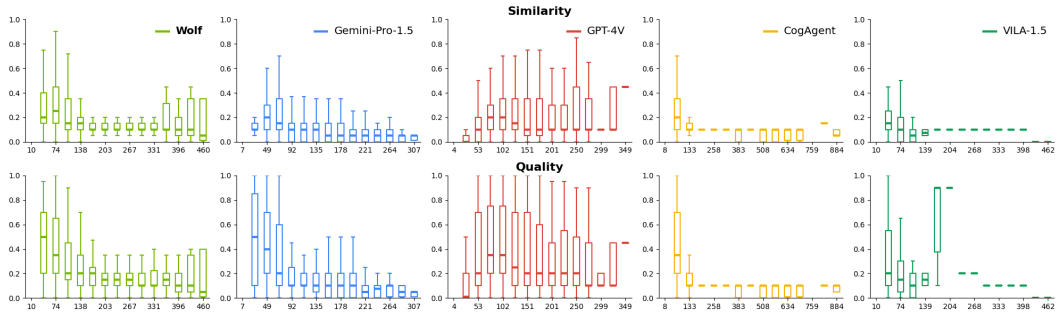


Figure 8: Caption Similarity / Quality evaluated by GPT-4 under varying caption length.

A.6 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 [14] and VILA [21] supporting code is heavily based on loading PIL images to present to a huggingface pipeline [41]. 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 [2] 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 [20, 11, 23] 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.

A.7 Author Statement

We confirm that all our annotated captions are under a CC License. We bear all responsibility in case of rights violations, etc. All of our results are easily reproducible. We will host and continue maintain the entire Wolf framework, metric, dataset, and benchmark.

A.8 Updated Results and Documentation

We will regularly update Wolf results and documentation on our webpage.