CALVIN: Improved Contextual Video Captioning via Instruction Tuning

Gowthami Somepalli[⋆♥], **Arkabandhu Chowdhury** [♠], **Ronen Basri** [♣], **Jonas Geiping** [♦], **Tom Goldstein** [♥], **David Jacobs**[♥]

[♥] University of Maryland, College Park [♠] Meta AI

[♦] ELLIS Institute, MPI for Intelligent Systems [♣] Weizmann Institute of Science

Abstract

The recent emergence of powerful Vision-Language models (VLMs) has significantly improved image captioning. Some of these models are extended to caption videos as well. However, their capabilities to understand complex scenes are limited, and the descriptions they provide for scenes tend to be overly verbose and focused on the superficial appearance of objects. Scene descriptions, especially in movies, require a deeper contextual understanding unlike general-purpose video captioning. To address this challenge, we propose a model, CALVIN, a specialized video LLM that leverages previous movie context to generate fully "contextual" scene descriptions. To achieve this, we train our model on a suite of tasks that integrate both image-based question-answering and video captioning within a unified framework, before applying instruction tuning to refine the model's ability to provide scene captions. Lastly, we observe that our model responds well to prompt engineering and few-shot in-context learning techniques, enabling the user to adapt it to any new movie with very little additional annotation.

Ground Truth: Lenihan knocks it backwards into the pool

VideoLlama: First, the video shows a soldier holding a rifle and standing on stairs. Next, a soldier holding a gun is seen standing on a rooftop in the rain.....the soldier is holding a bow and arrow and a rifle.

VideoLlaVa: The video shows a man in a military uniform holding a rifle and aiming it at a target. He then fires the rifle.Here are several guns visible in the image, including a rifle held by a soldier, a handgun, and a few other guns scattered around the area.

MovieChat: The video shows various scenes of an army man in a military setting, including a helicopter flying, a soldier standing in front of a building, a soldier standing in front of a tree, and soldiers walking

CALVIN (Ours): Lenihan shoots the alien in the head

Figure 1: A scene from MAD [80]-eval split. We present the captions generated by our model, represented as CALVIN against various off-the-shelf LLMs, with hallucinations highlighted in red. First, our model utilizes the context well, by understanding the name of the character is 'Lenihan' and that there is an alien in the scene, and second, our model has less hallucination and verbosity compared to other models.

⋆ Work done during internship at Meta. Correspondence: gowthami@umd.edu

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

1 Introduction

The volume of video data on the internet is increasing every year, now representing the largest portion of internet traffic [2]. To make this wealth of visual content accessible to vision impaired individuals requires audio descriptions. Audio descriptions (AD) describe and narrate videos or segments of videos in natural language, but are often still manually curated for only a few select videos.

The emergence of large, multi-modal vision-language models (vLLMs) has led to exciting progress on supplementing AD with automatically generated descriptions (AAD), but current models still suffer from notable weaknesses in videos. Consider the concrete example of a video of a woman smiling and waving at a bus carrying a friend. An observer with understanding of the context of the scene, like a human, might caption it as `A happy woman waving at the bus, perhaps receiving a loved one`, while a vLLM trained on image-caption pairs would provide a more literal description, such as `A woman wearing a red dress. A woman is smiling. A bus stopped. The scene has people in the background`. This illustrates a key shortfall: vision LLMs focus on superficial details like the properties of physical objects, which are often irrelevant to the broader context, because they are trained on static images. Several of these models process videos only as individual images, simply stringing together descriptions from each image while failing to capture the overarching narrative.

Secondly, current systems struggle to incorporate prior context into their description. Using the same scenario, if we inform a human that the woman's name is Mary and her husband, a soldier, is returning from war, a human would caption it as, `Mary is overflowing with joy to receive her husband, a war hero, at the bus stop`. Here, the human might choose to omit less significant details like 'waving' in favor of emphasizing her happiness, showing a nuanced understanding of the context. In contrast, many vision LLMs lack the ability to integrate such context with visual data, often ignoring it entirely. This is a significant gap, as useful interpretations naturally prioritize the intentions of characters and the results of their interactions through time, rather than the mere presence of objects and their movements - which would make the video hard to follow for a listener reliant on AD.

Drawing inspiration from the way humans utilize context in captioning, we introduce a novel contextual captioning model, CALVIN, that is designed and instruction-tuned to generate audio descriptions. Our primary objective is to develop a model that, given appropriate context, can generate captions closely resembling those crafted by humans. To accomplish this, we train our model using the Movie Audio Descriptions(MAD) dataset [80], which includes human-generated annotations for movie scenes, complete with timestamps. This enables us to construct a "text context" for each scene, based on preceding scenes. However, the MAD dataset alone is limited in scope and insufficient for fully training a video-LLM. To address this, we incorporate image-VQA datasets, which significantly enhance the model's visual understanding. We provide a detailed discussion on training methodologies, including the optimal combinations of data to use at different stages.

In addition to producing scene descriptions that are more human-like and useful than the existing vision models, CALVIN, achieves state-of-the-art (SOTA) performance in captioning on the MAD-eval dataset, with major improvements ($\sim$26% improvement on CIDEr and $\sim$68% improvement on BertScore) over the previously established SOTA model. CALVIN also performs significantly better than all the recent off-the-shelf video LLMs we studied on zero-shot evaluations on the TVC dataset [43]. We illustrate the model's capabilities in Figure 1. Here, while most video-LLMs generically describe the scene as `a soldier holding a rifle` or `an army man in a military setting` missing the narrative nuance of the ground truth caption `Lenihan knocks it backward into the pool`. CALVIN stands out by not only recognizing the character as 'Lenihan' but also incorporating prior information about the presence of an alien. Unlike the overly verbose captions of other models, our captions are crisp and narrative, focusing on actions and outcomes that are important to the plot, underlining their usefulness for the task of automated audio description.

We also introduce two test-time adaptation strategies, prompting and few-shot tuning, that are particularly useful in scenarios where additional contextual information is lacking. Finally, we describe some of the limits of our current model and outline potential avenues for future research to further advance this field.

2 Related Work

Video Understanding. The key objective of parsing spatiotemporal information in videos can be achieved through hand-crafted features [15, 18, 22, 69], recurrent networks [19, 36, 107], convolutional networks [24, 27, 51, 70, 90, 95], and more recently Vision Transformers (ViTs) [7, 9, 21, 23]. ViTs [21] treat an image as a set of patches and use a transformer architecture to model their interactions. Some works also effectively add multi-scale hierarchies [23, 28, 60, 73, 106] or local structures [14, 20, 60] to the transformers. Naturally these models can be extended to videos where a video is treated as a sequence of independent image frames, and a subsequent temporal pooling layer or a temporal transformer is added [7, 16]. There have been more generalized video modeling approaches [9, 23, 66, 68, 73, 84] that directly work on a video clip by dividing it into 3D spatio-temporal patches. While most of these models have proven to work well on short videos (<5 seconds), longer-video (>30 seconds) understanding is still an active research area. Some existing methods include pre-computing features and separately training backbones [3, 19, 26, 94], increasing model efficiency to include more frames [35, 38, 95, 110], and building a memory-model that can reference prior context [13, 41, 42, 96, 97]. Recent studies show that video-text pretraining [4, 5, 25, 29, 37, 45, 46, 49, 63, 71, 75, 82, 88, 89, 101, 103] can greatly help in long video understanding [83], temporal localization tasks [12, 44, 55, 100], text-video retrieval [63], video question answering [104], and video clip captioning [5, 75].

Generalist Video LLMs. Recent breakthroughs in language modeling have also spurred a flurry of work incorporating first image data, and then video data, as additional input modalities [5, 61]. While some works try to train large-scale video-text models directly from scratch [47, 65, 67, 87], most work focuses on continued pretraining and finetuning of base language models [102, 108]. Models may be generic multi-modal models [98, 105], or branch off from existing image-text models, such as LLaVA [52], targeting video understanding [54], and initial work on long video understanding [54]. Instruction tuning for videos was considered in [58, 59] and [78], and, with a particular emphasis on interactions with long(er) videos in [48, 81]. Understanding long videos is not only a learning problem, but also a technical challenge, necessitating engineering improvements, such as RingAttention to even process long videos [57].

Contextual Video Captioning. What are use cases for these video LLMs? A particularly interesting one is automated video captioning, which effectively converts video content back into text. This is useful, not only for tasks such as search, retrieval, but also essential to generate audio descriptions (AD) of video content. Automated audio descriptions describe the content of videos verbally, and are central to making videos accessible to anyone with visual impairment [64, 76, 93]. Seminal work, such as CineAD [11], argued that automated audio descriptions of even highly contextual content such as cinema, should be feasible, leading to a broader interest in automated audio descriptions (AAD). The most recent work in this direction are AutoAD I and II [30, 31], generating audio descriptons based on CLIP features processed with smaller language models. Auto-AD III [32] is a concurrent work that introduces a large-scale dataset and a 7b model to perform this task. AD is particularly focused on contextual descriptions and allows a listener to make sense of a long-form video, such as a movie. This separates this task from the more generic task of dense video captioning [33, 39, 77, 82, 91].

3 CALVIN: A contextual video captioner

We now discuss the proposed architecture, datasets, and training process.

3.1 Method & Architecture

Consider a collection, $\mathcal{V}$, of movie scenes $[\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_n}]$. Each scene contains k image frames such that $\mathbf{v_j} = [v_j^1, \ldots, v_j^k]$, and comes with annotations $\mathbf{v}_j$ in the form of a sequence of text tokens $[a_j^1, a_j^2, .., a_j^m]$. These annotations are usually obtained by converting audio to text, where the audio comes from the movie dialog or a person describing the scene. We seek a model $f_\phi(\cdot)$ that effectively predicts the audio description given the past visual and text tokens.

We parameterize $f_\phi(\cdot)$ with an LLM. While the pre-trained LLM can easily handle all the text tokens $\mathbf{a_{1:i-1}}$, it must be adapted to handle vision tokens corresponding to frames $\mathbf{v_i}$. We construct a video projection layer to process the video component and project it into the LLM's representation space. The overall causal problem formulation of contextual video captioning becomes

$$\hat{\mathbf{a}}_\mathbf{j}^\mathbf{i} = f_\phi\left(g_\theta(\mathbf{I}(v_j^1), \ldots, \mathbf{I}(v_j^k)), \mathbf{a_j^{1:i-1}}\right) \tag{1}$$

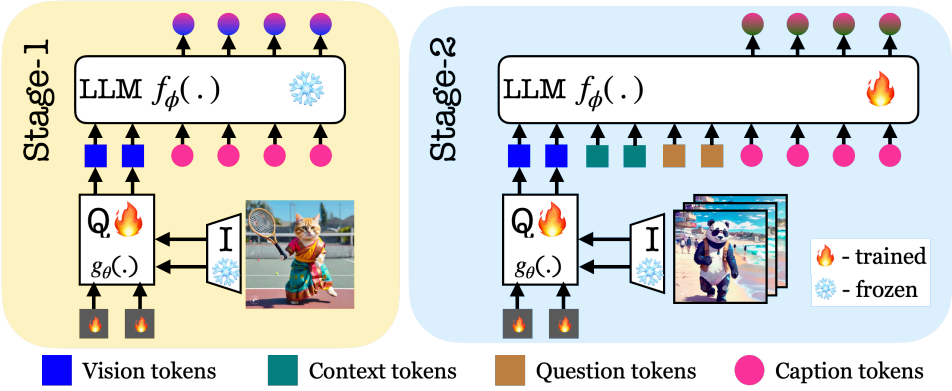

Figure 2: **CALVIN**: The architecture has 3 main components. (1) A frozen image embedding extractor I, (2) Non-linear projection module Q, and (3) An LLM. We train the model in 2 stages. Stage 1, we train only the projection module Q on image caption data. Stage 2, we use instruction formatted higher quality image-video data and finetune the parameters of Q and LLM. Refer to Sec. 3 for more details. Image and video examples shown here are synthetically generated using Meta Imagine [1].

Data format for training:
Stage 1: `<visual><caption>`
Stage 2: `<visual> <context> <question> <caption>`

Examples:
Stage 1: `[IMAGE_TOKENS] A cat playing tennis in a saree.`
Stage 2: `[VIDEO_TOKENS] Leonard the panda has a day off. He went for a swim. How would you describe the key visual elements and actions in this video? Leonard is taking a stroll down the beach`

Figure 3: **Stage-wise data format example**. We present the templates and an example of data format for different stages as shown in Figure 2. The colors of the text in this example match the token type colors in Fig. 2. The underlined text represents the segments that the language model must predict at all stages, and it is upon these predictions that the loss is computed.

where **I** is a *frozen* image embedding model that accepts an image frame v_i^k and outputs a fixed $d-$dimensional embedding. The function $g_\theta(\cdot)$ is a *learned* non-linear projection module that takes in multiple image embeddings and projects them into the LLM latent space. We use a CLIP ViT-h/14 [99] vision encoder as the frozen feature extractor **I**.

We train the system to predict the next text token based on all available context in the scene:

$$\theta^*, \phi^* = \arg\min_{\theta,\phi} \sum_{j=1}^{n} \sum_{i=1}^{m} \mathcal{L}\left(\hat{\mathbf{a}}_\mathbf{j}^\mathbf{i}, a_j^i\right)$$

where $\hat{\mathbf{a}}_\mathbf{i}$ is as defined in Equation (1) and $\mathcal{L}$ is cross-entropy loss.

Our projection module $g_\theta(\cdot)$ is comprised of two sub-components: a Q-Former layer [47], and a linear layer. The Q-Former begins with a fixed number of learnable embeddings, which are cross-attended by the video-frame embeddings during training. This formulation provides flexibility in handling a variable number of CLIP embeddings, up to a maximum defined by the position embedding, hence allowing training with mixed image and video datasets. The Q-Former layer is initialized with the pre-trained weights of BERT base [17]. We add position embeddings to the video-frame embeddings before the cross-attention to capture the temporal dynamics in videos. The second part of the projection module is a linear layer that projects the output of the Q-Former into the latent space of the LLM. We use a Llama-2 7b variant [85] model as our base LLM module.

3.2 Data and Stage-wise Training

The MAD dataset [80] stands out as the only extensive audio-description dataset currently available that includes aligned visual data, making it uniquely suitable for training contextual captioning

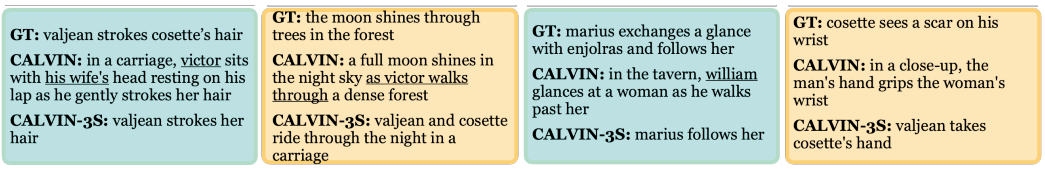

Figure 4: Qualitative results: Captions based on previous context are denoted **CALVIN-3S**. Captions without any context are represented as **CALVIN**. Contextual captions are close to ground truth and can get the names right based on the context. Without context, we see hallucinations (underlined), especially with the names.

models. However, the dataset's scale is relatively modest, especially when considering the extensive data requirements for effectively training a video-based LLM. To address this limitation, we train jointly on both image and video data. We use a single unified CLIP and Q-former vision pipeline for both modalities, enabling us to seamlessly transfer knowledge from the image to the video domain. We use a two-stage training process for the video LLM, as illustrated in Fig. 2.

In the first stage, we align the projection component (as denoted by **Q** in Figure 2), which consists of the Q-Former and a linear layer, while keeping both the CLIP and LLM frozen. In this stage, we use an internal dataset of image-caption pairs ($\approx$ 100M pairs), CC-12M [76] and the pretraining stage data curated by LLaVA [58, 59] which is built on top of MS-COCO [56].

For the second stage of training, we fine-tune the projection component as well as the LLM as shown in Fig. 2. This stage also marks our transition to employing higher-quality vision datasets. The data mixture for this phase is significantly enriched, comprising CC-3M [76], instruction tuning data from LLaVA 1.5 [58] – itself a curation based on a range of public image-caption or QA datasets including OKVQA [62], A-OKVQA [74], TextCaps [79], Visual Genome [40]. Additionally, we incorporate the WebVid-3M [8] video caption dataset and MAD [80] dataset train split into this stage. As such, this data mix is of higher quality as the majority of it is human-annotated.

Contrary to the approach of AutoAD [30] where the MAD dataset was trained separately, our ablation studies, which will be discussed later, reveal that integrating the MAD dataset into the second stage speeds up and simplifies training without significantly impacting the performance of the final model.

Turning Stage-2 data into instruction data: The dataset from LLaVA-1.5 comes pre-formatted for instruction tuning, simplifying its integration into our training process. For the other datasets, we adapt them to align with this instruction format. Typically, these datasets follow a `<image/video><caption>` structure. We restructure them into a more comprehensive format: `<image/video><context><question><caption>`. In this format, for all datasets except MAD, the `<context>` component is simply a placeholder space. For MAD, we use the annotation from the previous few scenes as `<context>`. We sample from a curated set of template questions regarding the `<question>` component. For a detailed view of these template questions, please refer to the Appendix.

4 Experiments & Results

Training details: All models are trained on a single A100 node with 8 GPUs. In Stage 1, we train only the projection module (Q-Former and linear layer) for $400,000$ iterations, with gradient accumulation over 4 steps and per-GPU batch size of 32. The learning rate has a cosine schedule and a warmup phase of 2,500 steps with min LR $1e-6$ and max LR $1e-4$. In Stage-2 we train Q-Former, linear projection, and the LLM. We train each model for $120,000$ iterations with a cosine learning rate with min LR of $1e-6$ and max LR of $1e-4$. The per-GPU batch size is 12 for image datasets and 6 for video datasets. Across all stages, a weight decay of 0.05 was applied. We adopt the Low-Rank Adaptation (LoRA) [34] approach for training the LLM. We set the LoRA rank to 32 and use LoRA on the QKVO (Query, Key, Value, and Output) in the attention layers. Unless stated, the Q-Former contains 4 layers with 32 learned embeddings and is trained to accept a maximum of 32 frames from the video. During the training, we sample 32 frames from the video. The datasets used in each stage are discussed in detail in Sec. 3.

Metrics: We evaluate our models and compare against baselines on four metrics. The first is BertScore [109], which measures the similarity in BERT representations between the ground truth and the generated captions. The remaining three are traditional captioning evaluation metrics, CIDEr [86], ROUGE-L [53] and, SPICE [6] each offering a unique evaluation perspective.

Evaluation details: We benchmark against the current state-of-the-art model, AutoAD-II [31], using the MAD-named-test split. This test split has captions that include character names. Owing to

the unavailability of trained models for both Auto-AD [30] and AutoAD-II [31], we present the best-reported results from these papers. Other baselines include ClipCap [65] and CapDec [67] which were discussed in Auto-AD I.

A limitation of the MAD dataset [80] is the absence of raw videos; only embeddings of the `CLS` token of the ViT model are shared. This constraint prevents the evaluation of models that require data in formats other than these specific embeddings. To provide a broader assessment of our model's performance relative to general-purpose open-source video language models, such as MovieChat [81], VideoLLaVA [52] and VideoLlama [108], we have conduct zero-shot comparisons on the TV-captioning (TVC) dataset [43].

4.1 Results

MAD-eval. Our main objective is to do contextual captioning. Hence we compare CALVIN against the SOTA models on the MAD-eval dataset in Tab. 1. Across all the metrics, we see CALVIN has significantly higher performance compared to the SOTA model. Even though our model is trained with context, we present a baseline case where we evaluate without context. In this case, we see $\sim$90% improvement over the best baseline BertScore. Among models with context, we see a further $\sim$26% improvement on CIDEr and $\sim$68% improvement on BertScore. We present a few qualitative results in Figure 4. CALVIN-3S represents the case with 3-scene context and CALVIN is the evaluation with no context. When evaluated with no context, the model hallucinates names in some scenarios (scenes- a, c). However, these hallucinations disappear once we provide context to the model and the captions are quite close to the ground truth. For scene b, our model's generation differs from the ground-truth caption but is technically correct. We encounter a few such false negative cases where a human would find this generation acceptable but it is hard to identify such cases without human intervention. We present some such scenarios in the Appendix.

TV-Captioning. We also benchmark CALVIN against off-the shelf SOTA video-LLMs on the TVC captioning [43] task in Table 2. All models are evaluated in zero-shot fashion, where none of the models (including ours) are fine-tuned on this dataset. Context is not available in this dataset, but this dataset's captions refer to the characters in the scene with names, and so we add the character names in the prompt when we are querying all the models. CALVIN outperforms all the baselines across all metrics despite not being trained on this dataset. The distinction is quite visible on the CIDER metric where CALVIN is $\sim$3x better than the next best model. This shows the generality of our model for movie/TV captioning tasks. We discuss a few ways of contextualizing captioning when the context data is not available in Section 5.

Table 1: **Evaluation on MAD-named-eval split.** The top half represents models evaluated without context. The bottom half shows the models trained/evaluated with context. Context column - the numbers in brackets show the number of scenes used in context. †- The numbers are from the original papers as the models are not public.

Model	Context	BertScore ↑	CIDER ↑	ROUGE-L ↑	SPICE ↑
ClipCap [65]	No	11.8	4.4	8.5	1.1
CapDec [67]	No	14.3	6.7	8.2	1.4
CALVIN (Ours)	No	**27.25**	**14.74**	**11.9**	**3.89**
AutoAD-I [30]†	Yes (6S)	23.8	21.9	13.9	4.8
AutoAD-II [31]†	Yes (Char.)	-	19.5	13.4	-
CALVIN (Ours)	Yes (3S)	39.08	25.47	16.30	7.33
CALVIN (Ours)	Yes (5S)	**40.18**	**27.71**	**16.83**	**7.76**

Table 2: **Zero-Shot evaluation on TV-Caption dataset.** All the models are provided with the names of characters in the scene. All the models use 7B LLMs.

Model	Zero-Shot	BertScore ↑	CIDER ↑	ROUGE-L ↑	SPICE ↑
VideoLlama [108]	Yes	28.29	3.34	5.61	3.52
MovieChat [81]	Yes	38.11	8.43	12.09	9.21
VideoLLaVA [52]	Yes	48.44	12.4	15.8	10.9
CALVIN (Ours)	Yes	**52.16**	**38.9**	**20.1**	**14.27**

Table 3: **Model Ablations:** Unless otherwise stated, all models are trained for the same number of iterations and on the same dataset. Unless otherwise stated, all the models are evaluated with a 3-scene context on MAD-named-eval split. ★-refers to the CALVIN 7B variant which we discuss throughout the paper

Ablation type	Config.	BertScore ↑	CIDER ↑	ROUGE-L ↑	SPICE ↑
Stage-2 Data	All ★	39.08	25.47	16.30	7.33
	(-) MAD	17.94	3.52	7.26	1.44
	(-) WebVideo	37.35	22.07	15.5	6.7
	(-) Image VQA data	34.99	20.21	14.53	6.3
	(-) All stage-2 data	16.20	2.24	7.14	1.24
Component finetuning	Q-Former	33.59	15.91	14.22	5.28
	LLM	11.18	0.1	0.4	0.3
	Q-Former + LLM ★	39.08	25.47	16.30	7.33
Q-Former tokens	32 tokens, width=4 ★	39.08	25.47	16.30	7.33
	64 tokens, width=4	38.83	24.27	16.22	6.94
	128 tokens, width=4	38.97	23.10	16.35	6.97
LLM tuning	LLM + prefix tuning	28.99	11.46	11.89	3.87
	LLM + Lora (rank=16)	38.63	24.67	16.08	7.09
	LLM + Lora (rank=32) ★	39.08	25.47	16.30	7.33
	LLM + Lora (rank=64)	39.27	24.73	16.57	7.18
Stagewise	MAD in Stage-2 ★	39.08	25.47	16.30	7.33
	MAD in Stage-3	39.34	23.46	16.34	7.05
Stage-2 Learning Rate	LR=$1e-5$ ★	39.08	25.47	16.30	7.33
	LR = $1e-4$	34.59	15.91	14.22	5.28

4.2 Ablations

We share a few ablations and some insights gained. Since there are many moving parts in the system, the search for our final configuration is mostly greedy and looks at one component at a time. We present the captioning metrics of the resulting models in Table 3. All models in the table are trained for the same number of iterations with a context of three previous scenes along with the video. These models are also evaluated with a 3-scene context, and other hyperparameters are kept constant.

Data mixture in Stage-2. As discussed in Section 3.2, we combine multiple datasets in this stage. We remove one data type at a time. Removing Stage-2 training impacts the performance the most, followed by removing the MAD dataset. This makes sense since AD captions tend to be crisp and more contextual, while others are more descriptive. There is a clear domain shift and we need the MAD dataset in training to bridge this gap. One interesting observation is that removing the image VQA data impacted the performance a bit more than removing the only video dataset other than MAD. This confirms that the image VQA data can contribute to video understanding.

Component tuning. As discussed in Section 3.2, we tune both Q-Former and LLM in Stage-2. We check performance when just one of these components is present. It turns out that training only the LLM leads to an extreme drop in performance while training only Q-Former dropped performance slightly, especially the CIDER score. We believe in LLM training case, the model completely ignores the vision embedding, attending only to the context, causing generated captions that are hallucinations with no grounding in the video. In the case of only Q-Former training, we believe that the model does not learn to utilize the context well, hence a slight degradation in performance.

Q-Former tokens. We examine the effect of the number of Q-Former tokens, which in turn controls the parameter count. More tokens causes a slight degradation in performance. We believe this is because the smaller size of the stage-2 dataset causes overfitting.

LLM tuning parameters. We examined two types of efficient LLM fine-tuning techniques - LoRA and Prefix Tuning [50]. Prefix-Tuning adds a few additional trainable parameters to each transformer block. In LoRA, a chosen set of parameters is updated by a low-rank approximation. First, we see LoRA training is significantly better than Prefix-Tuning. Second, as we increase the LoRA rank hyperparameter, we see slight performance improvement.

Other ablations. We depart from previous works by mixing the MAD dataset into stage II rather than separately fine-tuning on it. If we instead fine-tune on MAD with a 40,000 iteration stage III, we do not see much improvement. Additionally, we observe that reducing the max learning rate from $1e-4$ to $1-5$ in Stage 2 improves the performance by a non-trivial amount. We refer the reader to the Appendix for additional train-time and inference time ablations.

Prompt formats:
No context: [VISION_TOKENS]. Describe this visual.
Entity context: [VISION_TOKENS] The characters and entities in the scene are <ENT1> and <ENT2>. Describe this visual.

Example model outputs:
No context: She is talking to a blonde man.
Entity context: <ENT1> is talking to <ENT2>.

Figure 5: **Entity Prompting.** Instead of the previous scene context, we prompt the models with only the entity information. Refer to Section 5.1 for more details.

5 Video-language models are also few-shot learners!

It was shown in Brown et.al [10] that LLMs, trained on a diverse set of data, can benefit from prompt engineering and in-context learning. Despite our video representations being trained in the specific domain of video captioning, they still inherit the emergent properties of the parent LLM. This includes their ability to adapt to prompt engineering, in-context learning, and fine-tuning for specific tasks. In this section, we propose and evaluate two realistic strategies for customizing the model at test time. These strategies have shown improved performance in captioning compared to scenarios where no context is provided.

Regularization data: I don't recognize the characters or entities in the scene. Hence the final caption is: <MODIFIED_ORIGINAL_CAPTION>.

Test-time movie data: The characters and entities in the scene are <ENT1> and <ENT2>. Hence the final caption is: <ORIGINAL_CAPTION>.

Figure 6: **Few-shot fine-tuning**. We rewrite the captions in CoT [92] style and fine-tune the model on them.

5.1 Prompt engineering

It is hard for the models (and humans) to associate actors with their characters' names without a priori information. We observe that our model can be enhanced further by providing characters' or entities' information in the context. Assuming we can access entity information for a given scene, we prompt the model as illustrated in Figure 5. We noticed that just adding this information about entities in the scene to the context improves performance. See Section 5.3 for results.

5.2 Fewshot finetuning

Works like Dreambooth [72] "personalize" a large pre-trained model to an object or a category with test-time training. We test this idea by fine-tuning our model for every movie in the test set, showing it 20 scenes ($\sim$5 minutes) of each movie. This paradigm enables human-in-the-loop annotation where the humans can correct the model's mistakes while the model performs the bulk of the work.

To avoid over-fitting on the test time data, we propose to use additional regularization data from the train set (*i.e.*pre-annotated other movies) where we strip away the entity details using LLaMA-2 70B to get a rephrased caption without proper nouns or identifying details. See Figure 6 for an example of test-time movie data and regularization data used for test-time training. This is analogous to *chain-of-thought* prompting [92], where a model is trained to output its reasoning. In our use case, we show that with fine-tuning, the model recognizes some of the main characters as well as adapting to the tone of annotation for the movie. Note that annotators use different styles or languages based on the type of the movie. For instance, animated films have simplistic audio descriptions as they are directed at children. We share results in Section 5.3.

Experiments. We first sample 20 scenes from the test-time movie such that the scenes in which main characters first appear are included (see Table 4 for 50 instead of 20). Then we add the same number of samples for the regularization data, which are sampled randomly from the MAD-train split. We rephrase the captions of the whole mix in *chain-of-thought* (CoT) style as present in Figure 6. We start with CALVIN Stage-2 checkpoint and finetune it for 100 iterations with a constant learning rate of $1e-5$, 2-step gradient accumulation, and batch size of 10 on a single A6000 GPU. Each model training run takes less than 6 minutes.

GT: Lisa takes a few steps toward him and smiles.

CALVIN: She smiles at him and walks away.

CALVIN-E: She wears a tight-fitting black dress with a plunging neckline, and her blonde hair is styled in loose curls

CALVIN-FS: Lisa smiles at him.

GT: shaking his head, george shrugs his hands and smiles.

CALVIN: a smiling man wearing a blue suit and a pink tie walks down a hallway in an office building.

CALVIN-E: George smiles and shakes his head as he walks away.

CALVIN-FS: George smiles and nods.

Figure 7: **Qualitative results:** We present two adjacent scenes from one of the test movies. **CALVIN** corresponds to the result with no context. **CALVIN-E** refers to context containing only entity information. **CALVIN-FS** corresponds to the model few-shot trained on 20 random scenes from this movie (not the ones shown here.)

5.3 Results

We present the average metrics across all 10 movies in Tab. 4. We present results on CALVIN without context and CALVIN with entities in the prompt as discussed in Sec. 5.1. We present 2 fine-tuning scenarios trained with 20 to 50 samples from each test movie. Note that this accounts for less than 8% of total scenes with AD in all of the test movies.

We see performance improvement in both the personalization scenarios over the model without context. This shows that test time personalization can be a cheap and efficient way to improve contextual captioning. We present a qualitative example for 2 continuous scenes in Fig.7. The model without context often describes scenes in a generic albeit correct way. CALVIN with entities in the context is able to caption the scene a bit better, however, the model does not always use this information, e.g., the left scene caption does not refer to the character 'Lisa'. CALVIN-FS (fewshot) captions well with correct character names. See Appendix for ablations on the amount of training and number of few-shot examples. As a downside, few-shot finetuning seems to reduce caption diversity and length and occasionally associates wrong names with characters.

Table 4: **Test-time adaptation results:** (first row) CALVIN evaluation without context. (second row) 'Entities' means the context has just the list of entities in the scene as discussed in Sec. 5.1. (third and fourth rows) For few-shot training from Sec. 5.2, the number in brackets counts examples used in finetuning. Both strategies improve performance over the no-context.(† Numbers differ slightly from Tab. 1 since the numbers presented here are an average of metrics computed one movie at a time, and some metrics like CIDEr depend on the word distribution of evaluation set.)

Model variant	Context	BertScore ↑	CIDER ↑	ROUGE-L ↑	SPICE ↑
CALVIN †	None	26.91	10.85	11.76	3.73
Prompting §5.1					
CALVIN	Entities	39.14 (+12.23)	16.27 (+5.42)	15.56 (+3.8)	7.13 (+3.4)
Few-shot training §5.2					
CALVIN + FS (20)	None	34.14 (+7.23)	17.74 (+6.89)	14.06 (+2.3)	7.17 (+3.44)
CALVIN + FS (50)	None	36.19 (+9.28)	20.06 (+9.21)	15.22 (+3.46)	7.21 (+3.48)

6 Discussion and Conclusion

We present a state-of-the-art model for contextual scene captioning. Our model introduces several innovations, including instruction tuning and stage-wise training for various components, along with tailored data mixes at each stage. We demonstrate that our model, CALVIN, exhibits superior generalization capabilities and outperforms existing off-the-shelf video-LLMs at zero-shot evaluation on a TV-captioning dataset. We additionally propose novel test-time adaptation strategies involving prompting and few-shot tuning.

There are still ample opportunities for enhancement. These include processing longer videos, intelligent frame sampling for better and faster video representations, and audio processing. The biggest challenge to these advancements is data scarcity; There is a pressing need for public video datasets of high quality.

7 Acknowledgements

This work was made possible by National Science Foundation (NSF) grant #2213335. Further support was provided by the National Science Foundation (IIS-2212182), and by the NSF TRAILS Institute (2229885).

References

[1] Meta imagine. https://imagine.meta.com/. Accessed: 2024-01-10.

[2] Sandvine's 2023 global internet phenomena report. https://www.sandvine.com/press-releases/sandvines-2023-global-internet-phenomena-report-shows-24-jump-in-video-traf Accessed: 2024-02-01.

[3] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[4] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *Advances in Neural Information Processing Systems*, 34:24206–24221, 2021.

[5] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikoł aj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 23716–23736. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/960a172bc7fbf0177ccccbb411a7d800-Paper-Conference.pdf.

[6] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 382–398. Springer, 2016.

[7] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021.

[8] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1728–1738, 2021.

[9] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021.

[10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[11] Virginia P Campos, Tiago MU de Araújo, Guido L de Souza Filho, and Luiz MG Gonçalves. Cinead: a system for automated audio description script generation for the visually impaired. *Universal Access in the Information Society*, 19:99–111, 2020.

[12] Meng Cao, Tianyu Yang, Junwu Weng, Can Zhang, Jue Wang, and Yuexian Zou. Locvtp: Video-text pre-training for temporal localization. In *European Conference on Computer Vision*, pages 38–56. Springer, 2022.

[13] Yihong Chen, Yue Cao, Han Hu, and Liwei Wang. Memory enhanced global-local aggregation for video object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10337–10346, 2020.

[14] Zhengsu Chen, Lingxi Xie, Jianwei Niu, Xuefeng Liu, Longhui Wei, and Qi Tian. Visformer: The vision-friendly transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 589–598, 2021.

[15] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[16] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pages 7480–7512. PMLR, 2023.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[18] Piotr Dollár, Vincent Rabaud, Garrison Cottrell, and Serge Belongie. Behavior recognition via sparse spatio-temporal features. In *2005 IEEE international workshop on visual surveillance and performance evaluation of tracking and surveillance*, pages 65–72. IEEE, 2005.

[19] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[20] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12124–12134, 2022.

[21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[22] Efros, Berg, Mori, and Malik. Recognizing action at a distance. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 726–733. IEEE, 2003.

[23] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6824–6835, 2021.

[24] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.

[25] Yuying Ge, Yixiao Ge, Xihui Liu, Dian Li, Ying Shan, Xiaohu Qie, and Ping Luo. Bridging video-text retrieval with multiple choice questions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16167–16176, 2022.

[26] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 971–980, 2017.

[27] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 244–253, 2019.

[28] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12259–12269, 2021.

[29] Tengda Han, Weidi Xie, and Andrew Zisserman. Temporal alignment networks for long-term video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2906–2916, 2022.

[30] Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. Autoad: Movie description in context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18930–18940, 2023.

[31] Tengda Han, Max Bain, Arsha Nagrani, Gul Varol, Weidi Xie, and Andrew Zisserman. Autoad ii: The sequel-who, when, and what in movie audio description. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13645–13655, 2023.

[32] Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. Autoad iii: The prequel–back to the pixels. *arXiv preprint arXiv:2404.14412*, 2024.

[33] Mehrdad Hosseinzadeh and Yang Wang. Video captioning of future frames. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 980–989, January 2021.

[34] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[35] Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Timeception for complex action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 254–263, 2019.

[36] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2000–2009, 2019.

[37] Dohwan Ko, Joonmyung Choi, Juyeon Ko, Shinyeong Noh, Kyoung-Woon On, Eun-Sol Kim, and Hyunwoo J Kim. Video-text representation learning via differentiable weak temporal alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5016–5025, 2022.

[38] Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6232–6242, 2019.

[39] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715, 2017.

[40] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017.

[41] Sangho Lee, Jinyoung Sung, Youngjae Yu, and Gunhee Kim. A memory network approach for story-based temporal summarization of 360 videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1410–1419, 2018.

[42] Sangmin Lee, Hak Gu Kim, Dae Hwi Choi, Hyung-Il Kim, and Yong Man Ro. Video prediction recalling long-term motion context via memory alignment learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3054–3063, 2021.

[43] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, pages 447–463. Springer, 2020.

[44] Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34:11846–11858, 2021.

[45] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7331–7341, 2021.

[46] Dongxu Li, Junnan Li, Hongdong Li, Juan Carlos Niebles, and Steven CH Hoi. Align and prompt: Video-and-language pre-training with entity prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4953–4963, 2022.

[47] 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.

[48] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023.

[49] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. *arXiv preprint arXiv:2005.00200*, 2020.

[50] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

[51] Zhenyang Li, Kirill Gavrilyuk, Efstratios Gavves, Mihir Jain, and Cees GM Snoek. Videolstm convolves, attends and flows for action recognition. *Computer Vision and Image Understanding*, 166:41–50, 2018.

[52] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023.

[53] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[54] Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, Yumao Lu, and Lijuan Wang. Swinbert: End-to-end transformers with sparse attention for video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17949–17958, 2022.

[55] Kevin Qinghong Lin, Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Z XU, Difei Gao, Rong-Cheng Tu, Wenzhe Zhao, Weijie Kong, et al. Egocentric video-language pretraining. *Advances in Neural Information Processing Systems*, 35:7575–7586, 2022.

[56] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[57] Hao Liu, Wilson Yan, Matei Zaharia, and Pieter Abbeel. World model on million-length video and language with ringattention. *arXiv preprint arXiv:2402.08268*, 2024.

[58] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023.

[59] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.

[60] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[61] Chenyang Lyu, Minghao Wu, Longyue Wang, Xinting Huang, Bingshuai Liu, Zefeng Du, Shuming Shi, and Zhaopeng Tu. Macaw-llm: Multi-modal language modeling with image, audio, video, and text integration. *arXiv preprint arXiv:2306.09093*, 2023.

[62] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204, 2019.

[63] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9879–9889, 2020.

[64] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *2019 international conference on document analysis and recognition (ICDAR)*, pages 947–952. IEEE, 2019.

[65] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021.

[66] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3163–3172, 2021.

[67] David Nukrai, Ron Mokady, and Amir Globerson. Text-only training for image captioning using noise-injected clip. *arXiv preprint arXiv:2211.00575*, 2022.

[68] Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and Joao F Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. *Advances in neural information processing systems*, 34: 12493–12506, 2021.

[69] Xiaojiang Peng, Changqing Zou, Yu Qiao, and Qiang Peng. Action recognition with stacked fisher vectors. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 581–595. Springer, 2014.

[70] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *proceedings of the IEEE International Conference on Computer Vision*, pages 5533–5541, 2017.

[71] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[72] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023.

[73] Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, et al. Hiera: A hierarchical vision transformer without the bells-and-whistles. *arXiv preprint arXiv:2306.00989*, 2023.

[74] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *European Conference on Computer Vision*, pages 146–162. Springer, 2022.

[75] 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.

[76] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018.

[77] Zhiqiang Shen, Jianguo Li, Zhou Su, Minjun Li, Yurong Chen, Yu-Gang Jiang, and Xiangyang Xue. Weakly supervised dense video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1916–1924, 2017.

[78] Nina Shvetsova, Anna Kukleva, Xudong Hong, Christian Rupprecht, Bernt Schiele, and Hilde Kuehne. Howtocaption: Prompting llms to transform video annotations at scale. *arXiv preprint arXiv:2310.04900*, 2023.

[79] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioning with reading comprehension. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 742–758. Springer, 2020.

[80] Mattia Soldan, Alejandro Pardo, Juan León Alcázar, Fabian Caba, Chen Zhao, Silvio Giancola, and Bernard Ghanem. Mad: A scalable dataset for language grounding in videos from movie audio descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5026–5035, 2022.

[81] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Xun Guo, Tian Ye, Yan Lu, Jenq-Neng Hwang, et al. Moviechat: From dense token to sparse memory for long video understanding. *arXiv preprint arXiv:2307.16449*, 2023.

[82] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7464–7473, 2019.

[83] Yuchong Sun, Hongwei Xue, Ruihua Song, Bei Liu, Huan Yang, and Jianlong Fu. Long-form video-language pre-training with multimodal temporal contrastive learning. *Advances in neural information processing systems*, 35:38032–38045, 2022.

[84] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022.

[85] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[86] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[87] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022.

[88] Jinpeng Wang, Yixiao Ge, Guanyu Cai, Rui Yan, Xudong Lin, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Object-aware video-language pre-training for retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3313–3322, 2022.

[89] Jinpeng Wang, Yixiao Ge, Rui Yan, Yuying Ge, Kevin Qinghong Lin, Satoshi Tsutsui, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, et al. All in one: Exploring unified video-language pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6598–6608, 2023.

[90] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016.

[91] Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo. End-to-end dense video captioning with parallel decoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6847–6857, 2021.

[92] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[93] Spencer Whitehead, Heng Ji, Mohit Bansal, Shih-Fu Chang, and Clare Voss. Incorporating background knowledge into video description generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3992–4001, 2018.

[94] Chao-Yuan Wu and Philipp Krahenbuhl. Towards long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1884–1894, 2021.

[95] Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. Compressed video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6026–6035, 2018.

[96] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 284–293, 2019.

[97] Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13587–13597, 2022.

[98] Haiyang Xu, Qinghao Ye, Ming Yan, Yaya Shi, Jiabo Ye, Yuanhong Xu, Chenliang Li, Bin Bi, Qi Qian, Wei Wang, et al. mplug-2: A modularized multi-modal foundation model across text, image and video. *arXiv preprint arXiv:2302.00402*, 2023.

[99] Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying clip data. *arXiv preprint arXiv:2309.16671*, 2023.

[100] Mengmeng Xu, Erhan Gundogdu, Maksim Lapin, Bernard Ghanem, Michael Donoser, and Loris Bazzani. Contrastive language-action pre-training for temporal localization. *arXiv preprint arXiv:2204.12293*, 2022.

[101] Hongwei Xue, Tiankai Hang, Yanhong Zeng, Yuchong Sun, Bei Liu, Huan Yang, Jianlong Fu, and Baining Guo. Advancing high-resolution video-language representation with large-scale video transcriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5036–5045, 2022.

[102] Hongwei Xue, Yuchong Sun, Bei Liu, Jianlong Fu, Ruihua Song, Houqiang Li, and Jiebo Luo. Clip-vip: Adapting pre-trained image-text model to video-language representation alignment. *arXiv preprint arXiv:2209.06430*, 2022.

[103] Shen Yan, Tao Zhu, Zirui Wang, Yuan Cao, Mi Zhang, Soham Ghosh, Yonghui Wu, and Jiahui Yu. Video-text modeling with zero-shot transfer from contrastive captioners. *arXiv preprint arXiv:2212.04979*, 2022.

[104] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1686–1697, 2021.

[105] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023.

[106] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 558–567, 2021.

[107] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4694–4702, 2015.

[108] 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.

[109] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

[110] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of the European conference on computer vision (ECCV)*, pages 803–818, 2018.

CALVIN: Improved Contextual Video Captioning via Instruction Tuning

Supplementary Material

A Data curation additional details

All the human annotations in this paper are done exclusively by the authors. **Data Cleanup:** During the initial training of our models, we encountered an unexpected challenge. Despite using datasets generally regarded as high-quality, we observed that the models were outputting repetitive patterns of numerical data. A more detailed examination revealed that the captions in the WebVideo dataset often included specific dates or video resolutions, like 1930x1080 or 4HD. These elements were inadvertently leading the model to generate dates and resolutions in its output. We scrubbed numerical data from WebVideo using hand-crafted (regex) functions, resulting in a marked improvement in the quality of the model's output.

We also identified that the bounding box coordinate questions in the instruction fine-tuning dataset of LLaVA-1.5 [58], which are uncharacteristic of our problem domain, proved detrimental to the downstream captioning task. Hence, we excluded these.

Lastly, we noticed that MAD includes movie credits at the beginning or end of some films. We used LLaMA-70B [85] with a few human-annotated in-context examples to tag training examples as either credit-related or not. Removing scenes with credits caused a subtle improvement in the quality and relevance of the generated captions.

B Stage-2 training additional details

In Tab. 5, we present the question templates used in training of CALVIN Stage-2 model, to convert video-caption data into instruction data.

Table 5: Stage-2 question templates

Question template
What is this video about?
Describe the video, including the actions and scenes.
Provide a description of the given video, capturing its key moments.
Give a concise explanation of the video you see, including the events and characters.
Summarize the contents of the video, focusing on the main events and participants.
Detail the scenes and characters present in the provided video clip.
Narrate the sequence of events in the video, including significant actions.
Report on the events and individuals portrayed in the video clip.
Highlight the pivotal scenes and actions observed in the video.
What's going on in the video?
Describe the scene in the video
Summarize the key visuals and events of this video. What's happening in the video?
Detail the primary actions and visuals of this footage.
Provide a brief account of the scenes and characters within this video.
Elucidate the main events and visuals in this clip.
Give an overview of the storyline and characters present in this video.
Narrate the visual elements and the main events showcased in this video.
Chronicle the character dynamics and scene transitions in this video.
Describe the aesthetic elements and main occurrences in this footage.
Highlight the significant actions and interactions within this video.
Offer a concise breakdown of the scenes and events in this clip.
Detail the setting, characters, and major events of this video.
Explain the visual motifs and character roles within this video.
Decode the main story arc and visuals in this footage.
Provide an interpretation of the character relationships and visuals present.

C Qualitative analysis of CALVIN vs ground truth

In Tab. 6, we present a few examples of predicted caption vs ground truth caption, one from each movie in MAD-eval split. Clearly CALVIN-3S captions are as good as GT caps in many scenarios.

In the main paper Fig. 4(b), we have seen that the GT caption and CALVIN-3S captions differ quite a bit, however, the caption is an acceptable alternate when we looked at the video. We present a few cases from one of the MAD-eval movies, `HOW DO YOU KNOW?` in Tab. 7. This shows that the maximum achievable bert-score on this task is lower than 100 due to the subjective nature of this task.

Table 6: **Ground truth vs Predicted captions on CALVIN-3S model:** We present an example from each movie in the MAD-eval dataset.

Movie	Ground Truth Caption	Predicted Caption
IDES OF MARCH	stephen closes the door behind him	stephen shuts the door behind him
HOW DO YOU KNOW?	lisa presses the elevator button	lisa presses a button on the elevator's control panel
THE ROOMMATE	sarah smiles as stephen leans in and kisses her	sarah leans in and kisses stephen
BATTLE LOS ANGELES	joe nods at nantz	joe nods, then turns to nantz
Harry Potter and the goblet of fire	dumbledore thrusts it into harry hand	dumbledore hands it to harry
CHARLIE ST CLOUD	charlie eyes pool with tears	charlie's eyes well up with tears
LEGION	gabriel punches michael to the floor	michael punches gabriel in the face
LES MISERABLES	marius gently kisses eponine brow	marius and eponine kiss passionately
HANSEL GRETEL WITCH HUNTERS	hansel falls to the ground	hansel climbs down from the tree and collapses on the ground
SIGNS	graham stares blearily down at the floor	graham's gaze drops to the floor

Table 7: **Human acceptable predictions:** Some ground truth captions vs CALVIN-3S generated captions which the authors felt are acceptable despite being different from the GT. All the examples are from the MAD-eval movie, **How do you know?**

Ground Truth Caption	Predicted Caption
the elevator door shut	matty watches her go
george accepts the letter	she hands him the letter
lisa nods	lisa's mouth hangs open as she stares at him
in george office	the bespectacled man shakes his head
in his suit and dress shoes	george sprints down the sidewalk
annie loads the fridge	she sets the bag on the kitchen counter
now on the phone	george hangs up the phone

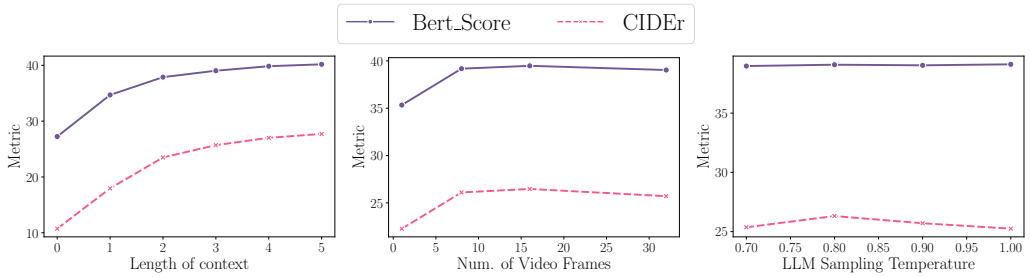

Figure 8: **Inference time ablations.** Effect of inference-time hyperparameters. (Left) Number of previous scenes in the context (Middle) Number of frames sampled per scene (Right) LLM sampling temperature.

D Additional ablations.

Inference-time ablations. In Figure 8, we show how BertScore and Cider metrics change with context length, number of video frames sampled, and LLM sampling temperature. We see that increasing the number of previous scenes in the context improves performance across both the metrics; The model was trained with just 3 scenes in the context and yet generalizes to more. Regarding the number of frames sampled from each clip, we see performance improvement as we go from 1 frame to more, indicating that the model uses motion information in caption generation. However, we see no improvement in performance beyond 8 frames. LLM sampling uses beam search. We do not see much difference in BertScore at lower temperature, but we see a slight bump in CIDEr.

E Fewshot-training ablations

We conducted an ablation on one movie from the MAD-train split - `3041 - JUST GO WITH IT` to understand the right number of samples and training iterations needed for few-shot experiments. We ablate the number of iterations in Fig. 9 and the number of annotated samples in Fig. 10. We observe that the performance does not improve beyond 50 annotated samples in training. And we see that training for more than 100 iterations does not improve the performance.

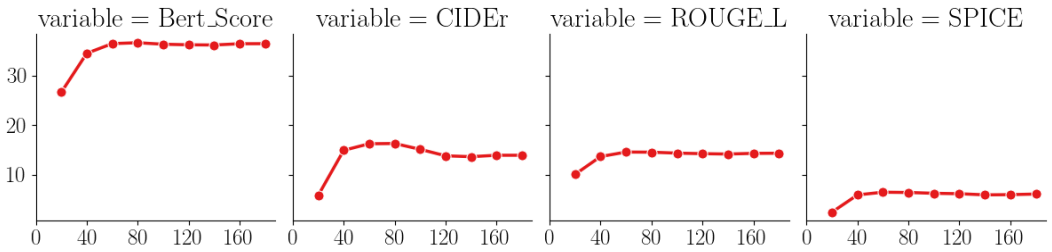

Figure 9: Number of training iterations vs Metrics.

F Broader Impacts.

Broader Impacts By leveraging previous scene contexts, CALVIN is well-positioned to enhance the accessibility of visual media for individuals with visual impairments, offering them a more immersive and contextually rich experience. This improvement could make entertainment and educational content more inclusive, promoting equal access to information and enjoyment regardless of visual capability. Furthermore, the technology could be applied in various other domains such as automated content moderation, where understanding the context of scenes can improve the detection of inappropriate content, and in digital humanities, where researchers can analyze films at scale to study cultural representations and evolution.

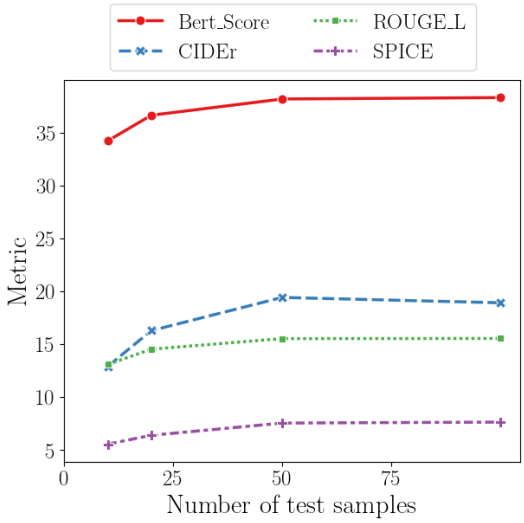

Figure 10: Number of ground-truth samples vs Metrics.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All the claims in the abstract and introduction are supported by empirical evidence presented in the Experiments section

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: We discuss the limitations of the method in the conclusion section.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We referenced all the publicly available datasets used in the training and explained in detail the model architecture and training details. We will release the code upon acceptance.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We referenced all the publicly available datasets used in the training and explained in detail the model architecture and training details.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We share the hyperparameters and ablations in the main and in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We could train models only once due to limited compute availability. We will add the statistical significance results for the camera-ready.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We discuss this in the experiments section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: To the best of our understanding, the paper conforms to the guidelines.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss this in the appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We discuss this in the appendix.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: To the best of our knowledge, all the assets used are properly credited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: We will release the assets if the paper is accepted.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.