

Robust Reasoning with Contextualized Visual Representation Learning

Anonymous submission

Abstract

Visual question answering (VQA) requires vision-language models (VLMs) to reason over images and respond to questions that ask about diverse details and inferences of these images. Typically, VLMs use pre-trained vision encoders to map visual inputs to feature representations and fuse these representations with large language models (LLMs), which generate responses to questions. However, these query-agnostic visual representations only reflect a static set of features of the visual input, which hinders VLMs from robustly responding to queries about out-of-distribution (OOD) features. To address this challenge, we propose fusing the query as additional context into the early-stage learning of vision encoding, which enables VLMs to learn context-aware visual representations that can flexibly adapt to different queries. Our Curriculum Vision-Context Fusion method, CVCF, learns the early integration of vision and context via a fine-grained curriculum learning scheme, based on a novel Contextual Vision-Inference Alignment (CVIA) dataset. We apply CVCF to two VLM architectures, and results on both demonstrate that CVCF effectively improves the reasoning robustness of VLMs, particularly when generalizing to OOD VQA data.

1 Introduction

Visual question answering (VQA; Singh et al. 2019) requires vision-language models (VLMs) to reason over visual inputs (images) with text-based queries, and generate appropriate textual responses. To answer visual questions, VLMs typically use pre-trained vision encoders to convert visual inputs to feature representations. These features are then concatenated with query embeddings, to enable reasoning across modalities via large language models (LLMs) (Liu et al. 2023a; Wang et al. 2024; Ghosh et al. 2024). However, the pre-trained vision encoder does not receive the queries as contexts when constructing the visual feature representations. As a result, current VLMs that rely on such context-agnostic representations may fail to generalize to out-of-distribution (OOD) queries (Mayilvahanan et al. 2024b; Liu et al. 2024b; Abbasi, Rohban, and Baghshah 2024), which may require visual representations that are more fine-grained than the ones yielded by a query-agnostic visual encoder.

For example, as shown in Figure 1, to answer question Q1, which asks about the season presented in the image, the

model must focus on the snow at the bottom part of the image. In contrast, for answering question Q2 that concerns the relationship between the creatures, the model needs to pay attention to the center of the image instead, where two creatures are engaged in a fight. VLMs with context-agnostic vision encoders such as CLIP (Radford et al. 2021) only capture visual features that are relevant to Q1, failing to extract features that are useful for answering Q2. This loss of visual information during early-stage encoding reduces the reasoning robustness of VLMs when encountering various real-world input queries.

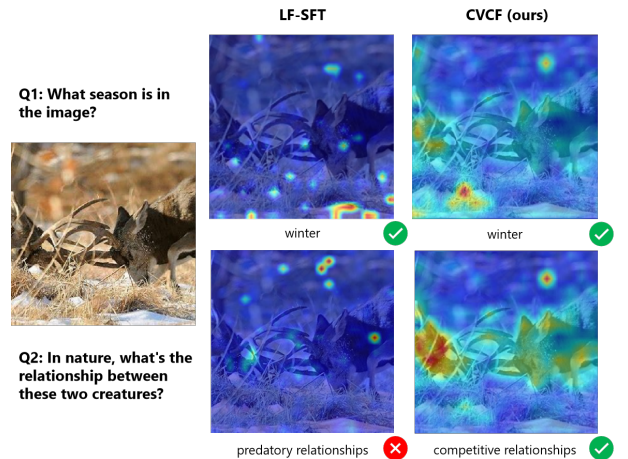


Figure 1: Illustration of how contextualized visual representation dynamically adapts to different contexts (queries). Heatmap areas with warmer colors indicate image regions that the model puts more attention to when responding to each query. LLaVA-1.5 architecture finetuned with our CVCF method shows more flexible attention to image regions that are relevant to each query.

In this work, we propose a Curriculum Vision-Context Fusion (CVCF) method to overcome the limitations of context-agnostic visual representations. CVCF guides the model to incorporate the input query (as additional context) into the early stages of vision encoding, enabling the learning of context-aware visual representations. These representations are dynamically tailored to be more relevant to varied

input queries, thereby enhancing the reasoning robustness of Vision-Language Models (VLMs), as shown in Figure 1.

To achieve this, CVCF proposed a three-stage curriculum learning pipeline designed to progressively enhance the alignment between the context-aware vision encoding and the corresponding output inference. The first two stages of this pipeline are powered by our novel **Contextual Vision-Inference Alignment (CVIA)** dataset, which we introduce along with its automatic generation pipeline. CVIA contains 10M image–context–inference triples, which are structured as contrastive learning samples across three difficulty levels (basic, easy, and hard), defined by the number of negatives and the difficulty of discriminating the positive sample. Specifically, in the first two stages of the pipeline, the early-fusion vision transformer is trained on CVIA, gradually progressing from the basic to the hard levels. And in the final stage, we fine-tune the entire VLM structure end-to-end on downstream VQA tasks to fully integrate the adapted vision encoding with the LLM’s inference process.

We evaluate CVCF by applying it into two leading VLM architectures: the encoder-decoder framework (Cho et al. 2021) and the decoder-only framework (Liu et al. 2024a). Across all benchmarks, models through our CVCF method consistently outperform the baseline learning methods, particularly in zero-shot settings. The results highlight CVCF’s robustness to domain shifts and its strong generalization to OOD contexts. This performance is attributed to CVCF’s teaching model to dynamically focus on contextually relevant regions of an image, as verified by our analysis.

2 Contextualized Visual Representation Learning

We propose Curriculum Vision-Context Fusion method, CVCF, designed to build context-aware visual representations from the early stages of visual encoding. Our method consists of three key components: (1) an early-fusion vision transformer that injects contextual information into the self-attention layers of the vision encoder, enabling rich cross-modal interactions; (2) a Contextual Vision–Inference Alignment dataset, constructed to support contrastive learning with controlled difficulty levels; and (3) a three-stage curriculum learning pipeline that progressively increases training complexity, promoting robust alignment between vision and context representations while enhancing generalization to downstream tasks.

2.1 Early-fusion Vision Encoding

To enable the model to effectively learn context-aware visual representations, we propose the usage of early-fusion vision transformer, which designed to early fuse the input context (query) with the visual features.

Unlike late-fusion approaches, which combine modalities only at LLM input stage, the early-fusion vision transformer injects context features into the self-attention layers of the vision encoder. This allows rich cross-modal interactions, leading to a context-aware representation that provides more effective feature extraction.

An early-fusion vision transformer consists of two main components: a vision encoder and a text encoder. The vision encoder is responsible for generating visual representations and the text encoder provides the contextual representations needed. Under such architecture, the vision transformer accepts both image and context as input, thereby shifting the representation from $V(I)$ to $V(I|C)$.

The input context C is encoded to produce textual features F_C via the text encoder. At each self-attention block of the vision encoder, these question features are concatenated with the visual sequence F_V , enabling a cross-modal attention over the combined sequence. The output corresponding to the visual sequence F'_{VC} is able to capture context-aware vision features:

$$F'_{VC} = \text{Attention}(\text{concat}(F_V, F_C))_{[0:M]} \quad (1)$$

where M is the length of vision sequence. Finally, the fused vision feature passes through both the original projection P and a learnable gated projection P_g , maintaining the layer’s outputs with minimal deviation at initialization while enabling a residual learnable stream of information:

$$F_{VC} = P(F'_{VC}) + P_g(F'_{VC}) \cdot \tanh(\beta) \quad (2)$$

The exact implementation of the early-fusion vision transformer can vary depending on the VLM architecture. For example, the text encoder can inherit the LLM’s encoder in an encoder-decoder architecture, while in a decoder-only setup, it can be initialized by a pretrained CLIP text encoder. Thanks to this flexibility, the early-fusion vision transformer can function as a plug-and-play module and has a wide applicability.

2.2 Contextual Vision-Inference Alignment Dataset

Recent findings (Li et al. 2024b) suggest that vision and language models learn similar representations of the world, differing only in spatial distribution. The cross-modality alignment mainly relies on the vision-language connector in a VLM. Since the early-fusion vision transformer modifies the original late-fusion (e.g. CLIP) architecture by introducing new input and parameters, the original output distribution is disrupted. As a result, we must retrain the model to learn the correct representations. To support this, we propose a **Contextual Vision-Inference Alignment (CVIA)** dataset, which contains three types of samples and supports curriculum contrastive learning.

The CVIA dataset categorizes samples into three difficulty levels: basic, easy, and hard. Basic samples are obtained via random negative sampling from the Cambrian dataset (Tong et al. 2024), while easy and hard samples are generated by our proposed data-augmentation strategies: negative image augmentation and negative context augmentation, respectively. An example is illustrated in Figure 2. We further denote our dataset as $\text{CVIA}_{\text{easy}}$ or $\text{CVIA}_{\text{hard}}$ when the majority of its samples come from the easy or hard level, respectively. Our design allows precise control over training difficulty by tuning the proportions of basic, easy, and hard samples. Detailed implementations of these strategies are provided in the following sections.





	Image	Context	Visual Inference
Positive		What is the condition of the sky in the image?	The sky in the image is cloudy.
Negative _{basic}		What is the main landmark in the image?	The Palace of Westminster.
Negative _{easy}		What is the condition of the sky in the image?	The sky is clear and blue outside.
Negative _{hard}		Which sign is positioned above the other?	The One Way sign.

Figure 2: An example from our CVIA. The dataset is constructed for curriculum contrastive learning, containing positive samples paired with negative samples across three difficulty levels: basic, easy, and hard. Each level includes ten negative samples per positive instance.

Negative Image Augmentation For each sample, our objective is to identify a different image that yields a different visual inference when paired with the same context. To achieve this, we search for negative images within the Cambrian dataset by computing CLIP embeddings and selecting those that have high cosine similarity scores to the positive image. We then apply a two-round verification with a VLM to ensure that the positive inference is not a plausible answer for the negative images, and vice versa. Finally, we employ a LLM to rewrite each negative inference to eliminate any length-based shortcuts.

We label the data produced by this strategy as "easy" negatives, since each positive sample can only be paired with exactly one negative in one contrastive batch, because there is no guarantee that different negatives will be mutually irrelevant. We include our detailed prompts for sampling negative images in Appendix.

Negative Context Augmentation Negative context augmentation aims to pair each image with a diverse set of contexts. Although the Cambrian dataset includes several contexts per image, we enrich this pool by following prior work (Zhu et al. 2024) and implementing an automated questioning pipeline to generate additional negative contexts and their corresponding inferences.

Our pipeline utilizes two separate VLMs: a questioner, which formulates questions about the image, and an answerer, which provides the visual inferences. The pipeline proceeds in three levels (coarse-grained, fine-grained, and reasoning-required) of question generation. The questioner moves forward to the next level once no new questions can be generated on the current topic. Additionally, we still deploy a LLM to complete the rewriting process to prevent shortcuts.

We label the resulting samples as "hard" negatives, since each image can be paired with multiple, mutually irrelevant contexts, forcing the model to learn stronger discrim-

inative capabilities when all negatives are included in the same contrastive batch. The detailed prompts for negative context sampling can be found in Appendix.

2.3 Curriculum Learning Pipeline

Curriculum learning (Wang, Chen, and Zhu 2021) introduces training samples in increasing order of difficulty, enabling models to first master simpler examples before tackling more complex ones. This gradual progression has been shown to improve convergence and final performance, particularly in transfer learning scenarios where smoother adaptation to new tasks is critical.

Inspired by this approach, we designed a three-stage curriculum learning pipeline (Figure 3) with two distinct training objectives: aligning visual features with contextual understanding and enhancing performance on downstream tasks. The first two stages focus on curriculum contrastive learning, training the model to align visual representations with their corresponding contextual inferences. In the final stage, we apply supervised finetuning to the entire VLM, allowing us to evaluate its effectiveness on downstream tasks.

We believe that this staged approach fosters a strong generalization capability, enabling the early-fusion vision transformer to function as a strong and flexible vision encoder when plugged into a VLM. The detailed strategy for each stage is described as follows.

Contrastive Learning Stages We introduce a novel contrastive learning objective that requires three inputs: image, context, and visual inference. The goal is to align the visual representation, conditioned on both the image and context, with the representation of its corresponding visual inference. We employ a two-stage training strategy that progressively increases data difficulty while gradually unfreezing model parameters, with a smooth transition from the pretrained CLIP encoder to our new objective.

Stage One In the first stage, the early-fusion vision transformer trains on CVIA_{easy}. At this stage, the text encoder is kept frozen and only the newly added vision encoder parameters are updated. This configuration enables the model to begin embedding contextual information into its visual representations while preserving CLIP’s foundational encoding capabilities. By starting with simpler examples and limiting the scope of parameter updates, we reduce the risk of overfitting or shortcut learning, and ensure a stable shift from CLIP’s original contrastive objectives to our context-aware targets.

Stage Two In the second stage, we train on CVIA_{hard}. Here, we unfreeze the text encoder so it can fully participate in the joint contrastive objective. This stage mainly focuses on distinguishing representations of the same image under different contexts, reinforcing the model’s ability to infer how context shapes visual interpretation. By gradually increasing both data difficulty and model flexibility, our two-stage contrastive curriculum enables the model to learn a rich, context-sensitive visual embedding, laying a robust foundation for subsequent downstream fine-tuning.

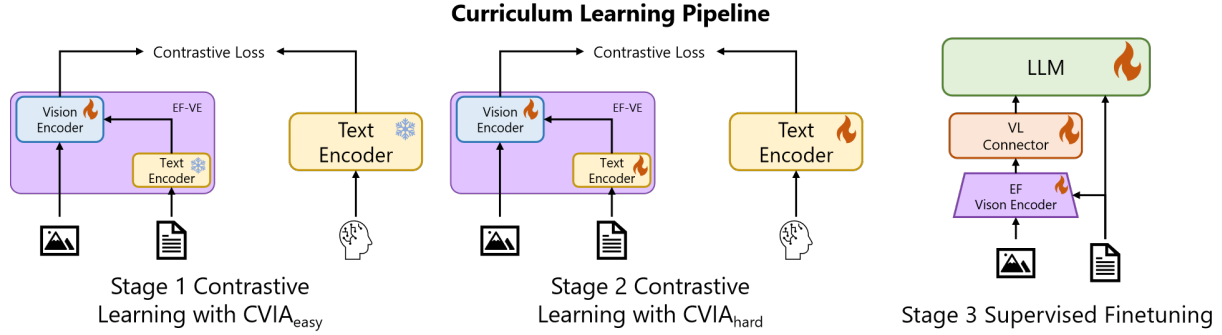


Figure 3: An overview of our three-stage curriculum learning pipeline. Stage 1 and Stage 2 use curriculum contrastive learning to finetune the early-fusion vision transformer using CVIA dataset under two difficulty settings. Stage 3 use supervised finetuning to finetune the entire VLM architecture. Note that EF-VE stands for Early-Fusion Vision Encoder.

Supervised Finetuning Stage In the third stage, we plug the early-fusion vision transformer into the VLM framework and perform supervised finetuning on downstream tasks. We follow the common VLM training paradigm, which first pre-trains the vision-language connector to align the vision representation with the text representing, and then finetunes the entire VLM model.

3 Experimental Setup

In this section, we detail the model configurations and experimental settings used in our data augmentation, training, and experiments.

3.1 VLM Finetuning Method

We evaluate our proposed method, CVCF, on two prominent types of VLM architectures: encoder-decoder and decoder-only. For the encoder-decoder setup, we adopt Flan-T5 (Chung et al. 2024) as the backbone language model and evaluate performance on four VQA benchmarks: TextVQA, VQA-v2 (Antol et al. 2015), ST-VQA (Biten et al. 2019), and VizWiz (Gurari et al. 2018). For the decoder-only setup, we follow the LLaVA-1.5 configuration and use Vicuna-1.5 (Zheng et al. 2023) as the language backbone. We evaluate performance on eight VQA benchmarks: TextVQA, VQA-v2, VizWiz, MME (Fu et al. 2023), POPE (Li et al. 2023b), MMBench (Liu et al. 2024c), ScienceQA (Lu et al. 2022), and SEED (Li et al. 2023a). For both architectures, we compare CVCF against two baseline training paradigms: (a) Late-Fusion Supervised Finetuning (LF-SFT), which uses pre-trained CLIP as the vision encoder, and directly plugs it into the VLM framework to perform SFT; (b) Early-Fusion Supervised Finetuning (EF-SFT), which replaces CLIP in SFT by the early-fusion vision transformer that applies attention to the context (but without contrastive learning).

3.2 Data Construction

We used some specific models in the process of building our CVIA dataset. We use MetaCLIP-h14 (Xu et al. 2024) as vision encoder to generate vision embeddings for similarity calculation. MiniCPM-V-2.6 (Yao et al. 2024) is used in both

data augmentation pipeline for image filtering and new context (query) generation. For rewriting the visual inference, we adopt Llama-3.1-70B (Grattafiori et al. 2024).

3.3 Training Settings

For the contrastive learning stages (Stages 1 and 2), we use 1M samples from our CVIA dataset for training, running for 10 epochs at each stage. The data distribution in CVIA_{easy} consists of 40% basic (random negative), 40% easy (image negative), and 20% hard (context negative) samples. In contrast, CVIA_{hard} is composed of 10% basic, 10% easy, and 80% hard samples, emphasizing more challenging instances.

In the supervised finetuning stage (Stage 3), CVCF employs different strategies depending on the underlying VLM architecture. For encoder-decoder architecture, we adopt LoRA (Hu et al. 2022) during training and finetune on a collection of datasets including TextVQA, COCO (Lin et al. 2014), DocVQA (Mathew, Karatzas, and Jawahar 2021), ChartVQA (Masry et al. 2022), VQA-v2. For decoder-only architecture, we follow LLaVA-1.5, using the same dataset composition as in its original setup.

4 Results

Method	TextVQA	VQA _{v2}	ST-VQA	VizWiz
LF-SFT	48.0	72.7	52.7	27.0
EF-SFT	49.7	73.2	54.5	27.6
CVCF (ours)	51.2	74.8	55.3	29.1

Table 1: Evaluation results on Flan-T5-XL encoder-decoder architecture. The vision encoders used in both EF-SFT and CVCF are initialized with CLIP-ViT encoder weights, while their text encoders are initialized as Flan-T5-XL encoder.

4.1 Evaluation on Downstream Tasks

We evaluate the effectiveness of our proposed CVCF by applying it within both encoder-decoder and decoder-only VLM architectures. We compare its performance against the Late-Fusion SFT (LF-SFT) baseline and the Early-Fusion SFT (EF-SFT) baseline across a range of benchmarks.

Method	TextVQA	VQA _{v2}	VizWiz	MME	POPE(rand/pop/adv)	MMBench	ScienceQA	SEED _{img}
LF-SFT	58.2	78.5	50.0	1510	87.3/86.1/84.2	64.3	66.8	66.1
EF-SFT	59.0	79.1	51.2	1518	88.0/87.2/85.3	65.1	67.0	66.2
CVCF (ours)	60.8	80.0	52.1	1533	89.1/87.6/86.1	66.2	68.9	67.2

Table 2: Evaluation results on LLaVA-1.5 decoder-only architecture. All methods employ Vicuna-1.5-7B as the initial LLM to be trained with the vision encoder. The vision encoders used in both EF-SFT and CVCF are initialized with CLIP-ViT encoder weights, while their text encoders are initialized with the weights of CLIP’s text encoder.

For the encoder-decoder architecture, we adopt Flan-T5 (Chung et al. 2024) as the base model and evaluate on four datasets, as shown in Table 1. For the decoder-only architecture, we use Vicuna-1.5 as the base model and compare our approach to LLaVA-1.5 following LLaVA evaluation benchmarks, including eight benchmarks (see Table 2).

Across both settings, CVCF consistently outperforms the baselines, demonstrating the effectiveness of learning contextualized vision representations through curriculum learning. Furthermore, when compared to other LF-SFT VLMs with comparable parameter sizes but different base models, CVCF achieves superior performance on nearly all datasets (see Appendix). This highlights the generality and robustness of our approach. Moreover, the design of CVIA and our curriculum learning pipeline positions CVCF as a general-purpose strategy for enhancing performance in vision-language tasks.

4.2 Generalization Ability on Zero-shot Datasets

Method	M ³ CoT	IllusionVQA Comp	S-L
LF-SFT	27.1	30.8	24.6
EF-SFT	35.5	31.6	23.3
CVCF (ours)	38.2	31.7	27.9

Table 3: Evaluation results on M³CoT (Chen et al. 2024a) and IllusionVQA (Shahgir et al. 2024) using the same setup as in Table 2. In IllusionVQA, "Comp" and "S-L" denote the Comprehension and Soft-Localization tasks respectively.

We hypothesize that directly fusing contextual information into the vision encoder and training it using our CVCF approach would yield more robust visual representations, thereby enhancing overall VLM performance. To validate this hypothesis, we evaluate our approach in a zero-shot setting on two recent benchmarks: M³CoT, which emphasizes the reasoning ability within the visual modality, and IllusionVQA, which assesses a model’s behavior on absurd or contradictory images. As shown in Table 3, our method significantly outperforms the baselines, most notably on the M³CoT benchmark and the Soft-Localization (S-L) task of IllusionVQA. These results demonstrate that context-aware vision encoding offers strong generalization ability, enabling the model to effectively handle OOD visual inputs.

5 Analysis

5.1 How curriculum learning effects the final performance?

Method	TextVQA	VizWiz
LF-SFT	40.2	23.7
CVCF	45.1	26.2
w/o curriculum (data)	44.4	24.4
w/o curriculum (freeze)	44.5	24.9
w/o curriculum (unfreeze)	44.7	25.5

Table 4: Accuracy on the TextVQA and VizWiz dataset with different contrastive learning strategy. The methods uses Flan-T5-base as the backbone language model.

Given a model and a dataset, there are often multiple strategies to adapt the model to a target task. In our experiments, we investigate the effect of curriculum learning and compare it against standard contrastive learning approaches (see Table 4).

We evaluated three different training configurations without curriculum strategy. In the w/o curriculum (**data**) setting, only samples labeled as hard in CVIA were used during training. In the w/o curriculum (**freeze**) setting, the text encoder parameters remained frozen throughout training. In contrast, for w/o curriculum (**unfreeze**), the text encoder was set to be fully trainable.

All three configurations lead to performance drops. The **data** setting causes the largest decline, suggesting that the absence of easier examples hinders the model’s ability to learn robust alignments. Both the **freeze** and **unfreeze** settings also negatively impact performance. We attribute this to different causes: freezing limits adaptation due to insufficient trainable parameters, while unfreezing risks diminishing pretrained knowledge after excessive parameter updates.

Overall, our experiments demonstrate that curriculum learning enables the vision encoder to better balance the retention of pretrained knowledge with the adaptation to new tasks, resulting in improved performance on downstream evaluations.

5.2 Ablation Study

In previous experiments, we utilized our CVIA dataset for finetuning the VLM in our CVCF method. Here, we investigated the role of our CVIA dataset by shifting the use of it from the contrastive learning stage to the supervised finetuning stage. As shown in Table 5, while this addition of

Method	TextVQA	VQA _{v2}
Flan-T5-XL		
w/ EF-SFT	49.7	73.2
w/ EF-SFT + CVIA	49.5	73.4
w/ CVCF	51.2	74.8
LLaVA-1.5		
w/ EF-SFT	59.0	79.1
w/ EF-SFT + CVIA	59.4	79.3
w/ CVCF	60.8	80.0

Table 5: Ablation study on the utilization of CVIA at different training stages. Our curriculum learning method CVCF uses CVIA data in the early-stage contrastive learning (Stage 1 and Stage 2), while the ablated baseline EF-SFT + CVIA adds our CVIA data (positive samples) into the late-stage supervised finetuning (Stage 3).

data improves over the baseline, it does not match the performance of our curriculum learning approach.

Notably, our curriculum learning pipeline is also significantly more efficient. By focusing on the vision encoder, which is much smaller than the entire VLM structure, our method reduces total training time by approximately one-third while achieving superior results. These results confirm that our proposed curriculum learning strategy is both more effective and more efficient.

5.3 Stratified Performance Analysis

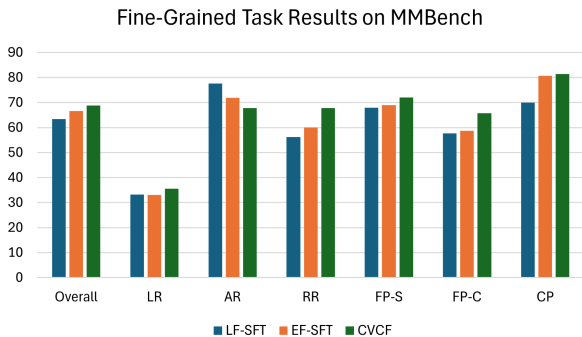


Figure 4: Comparison of accuracy on fine-grained tasks in MMBench. LR, AR, RR, FP-S, FP-C, and CP represent Logical Reasoning, Attribute Reasoning, Relation Reasoning, Fine-grained Perception (Single-instance), Fine-grained Perception (Cross-instance), and Coarse Perception, respectively.

In this section, we investigate deeper into the specific abilities that our CVCF method brings to a VLM. To this end, we evaluate the VLM on MMBench, which is a benchmark that contains six subsets that evaluate on different aspects of the multimodal understanding. Figure 4 shows the detailed performance histogram across these subsets.

As illustrated in the figure, CVCF achieves substantial improvements across all perception-related tasks, as well as in logical and relational reasoning tasks. These gains align with

the core design of our context-aware approach. By introducing contextual information into the vision encoder at an early stage, the encoder is better equipped to produce semantically rich and context-relevant features, ultimately leading to improved performance.

However, an exception is observed in the attribute reasoning subset, where CVCF demonstrates a drop in accuracy. Upon examining samples from this subset, we find that about 33% samples from this subset contain abstract or vague phrasing, such as "The object shown in this figure:", which lacks meaningful semantic content. As a result, CVCF struggles to effectively align the contextual information with the visual input. Therefore, this leads to the outcome that a well-trained VLM via our CVCF method underperforms compared to LF-SFT method.

5.4 Why do contextualized vision representations lead to accurate answers?

In this section, we investigate why VLMs tend to produce more accurate answers when using our CVCF to generate contextualized vision representations. Our hypothesis is that CVCF could effectively emphasize the relevant regions of an image that are critical for answering a given question or for supporting reasoning processes.

To investigate this hypothesis, we utilize Attention Rollout (Chefer, Gur, and Wolf 2021) to visualize the vision encoder’s attention through heatmaps. Attention Rollout computes a unified attention map by recursively aggregating attention matrices across all transformer layers. This process highlights the contribution of each input (image) token to the final output, offering insight into how contextual information guides visual reasoning and why our CVCF performs better than all baselines.



Figure 5: An example from MMBench. We compare attention heatmaps generated under the LLaVA-1.5 architecture using LF-SFT, EF-SFT, and CVCF method. In these heatmaps, the highlighted (red) regions indicate areas with greater influence on the model’s output. Only our CVCF method selects the correct answer.

Figure 5 well explains that after fusing contextual information, our CVCF significantly adjusts its self-attention patterns compared to context-agnostic image representations, even demonstrating a degree of visual reasoning ability. In

this example, the main objects in the image are a man and a Honda car. Without specific contextual guidance, it is natural for the LF-SFT method to focus on the man’s limbs, and even the car logo. However, these regions are not relevant to the correct answer. EF-SFT method, without being explicitly trained to fuse contextual information, focuses on the man’s center of gravity and the wooden floor behind him. Although these features are somewhat related to the question, it fails to reach the correct answer. As a result, it concludes that all situations are equally plausible.

In contrast, our CVCF demonstrates a markedly different attention pattern. Compared to the two baselines, the attention shifts significantly toward the wooden floor and the junction where it meets the stone road. This shift suggests that our method is aware of the height difference between the two surfaces. Remarkably, we can infer that the highlighted junction is the exact spot where the man’s right leg previously missed the step. With CVCF, the VLM focuses on the correct visual cues, enabling it to infer the reason for the man’s posture and movement, and to make the correct deduction that he is about to fall.

This example provides strong evidence that contextualized vision representations offer more effective visual information to VLMs. By aligning attention with semantically relevant regions, they enhance the model’s reasoning process and lead to more accurate answers in vision-language tasks.

6 Related Work

6.1 Contrastive Learning for Vision Models with Language

Contrastive learning (Hadsell, Chopra, and LeCun 2006) is a form of an unsupervised learning method that aims to learn data representation by maximizing the similarity between related samples and minimizing the similarity between unrelated samples.

Traditional vision models, such as convolutional neural networks (CNNs) (Szegedy et al. 2015; He et al. 2016) and Vision Transformers (ViTs) (Dosovitskiy et al. 2021), are typically limited to predicting predefined categories in classification tasks. Although many works (Chen et al. 2020; He et al. 2020) have attempted to enhance the capabilities of visual models through contrastive learning, CLIP (Radford et al. 2021) was the first to utilize it to align image and text modalities. CLIP consists of two parallel encoders for images and text. It learns a shared embedding space in which semantically aligned image-text pairs have high similarity scores. This multimodal alignment enables CLIP to generalize to novel visual concepts beyond the training categories (Mayilvahanan et al. 2024a).

In this work, we modified the CLIP framework to an early-fusion architecture so that the vision encoder can accept both vision and associated text inputs. Therefore, our contrastive learning objective is adapted to align the context-aware visual representation with its corresponding visual inference.

6.2 Enhanced Vision Representation

Traditional late-fusion architectures (Bai et al. 2025; Chen et al. 2024b) combine visual features and text embeddings at the stage just before the final LLM inference generation. With the late-fusion supervised finetuning, the vision encoder could only produce a context-agnostic image representation that related to image itself, constraining its ability to extract task-specific details and, consequently, hindering performance on complex vision-language tasks (Li et al. 2024a).

Many studies have explored strategies for improving visual representations (Li et al. 2021; Liu et al. 2023b; Vyskočil and Pícek 2023; Shi et al. 2024) to support more efficient image encoding. For example, BRAVE (Kar et al. 2024) let the model learn to generate its visual representation by employing k independent vision encoders and concatenating their outputs, which enables the extraction of visual features from multiple perspectives, thereby enriching the visual representation space. However, the model may still struggle (Geigle et al. 2023) when image-text inputs require features beyond the collective scope of these encoders.

Another solution is to convert the late-fusion in to an early-fusion supervised finetuning method, which introduce contextual information at the early stage of image feature processing, leading to dynamic vision representations that adapt to the accompanying context.

For instance, Chameleon (Team 2024) adopts an early-fusion method by treating images as discrete tokens like the text. It jointly embeds image and text tokens into an autoregressive language model, allowing direct integration of visual and linguistic features. Despite its elegance, this method suffers from information loss during tokenization (Fan et al. 2024), as discrete tokens can only capture a limited subspace of the continuous visual feature space. QA-ViT (Ganz et al. 2024) accomplishes the early-fusion by augmenting a frozen CLIP vision encoder to form a conditional encoder, enhancing the image representations by utilizing self-attention layers to integrate context information. However, it is only trained within the VLM architecture without individual training of vision encoding, limiting the model’s performance, as shown by the performance of EF-SFT baseline.

7 Conclusion

We propose CVCF, a curriculum vision-context fusion method that fuse context information into the early-stage of visual encoding, enabling the model to construct dynamic context-aware visual representations. To support effective learning of such contextual representations, our method introduce a fine-grained curriculum learning pipeline built upon our CVIA dataset. Comprehensive experiments on two distinct VLM architectures show that CVCF consistently enhances reasoning robustness and generalization, especially under out-of-distribution (OOD) settings. Thanks to its modular design, CVCF allows for easy integration into a wide range of VLMs, making it a general paradigm for enhancing VQA robustness across diverse scenarios.

References

- Abbasi, R.; Rohban, M. H.; and Baghshah, M. S. 2024. Deciphering the role of representation disentanglement: Investigating compositional generalization in clip models. In *European Conference on Computer Vision*, 35–50. Springer.
- Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C. L.; and Parikh, D. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, 2425–2433.
- Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; et al. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.
- Biten, A. F.; Tito, R.; Mafla, A.; Gomez, L.; Rusiñol, M.; Jawahar, C.; Valveny, E.; and Karatzas, D. 2019. Scene Text Visual Question Answering. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 4290–4300. IEEE.
- Chefer, H.; Gur, S.; and Wolf, L. 2021. Transformer Interpretability Beyond Attention Visualization. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 782–791. IEEE.
- Chen, Q.; Qin, L.; Zhang, J.; Chen, Z.; Xu, X.; and Che, W. 2024a. M³CoT: A Novel Benchmark for Multi-Domain Multi-step Multi-modal Chain-of-Thought. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 8199–8221. Bangkok, Thailand: Association for Computational Linguistics.
- Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.
- Chen, Z.; Wang, W.; Cao, Y.; Liu, Y.; Gao, Z.; Cui, E.; Zhu, J.; Ye, S.; Tian, H.; Liu, Z.; et al. 2024b. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.
- Cho, J.; Lei, J.; Tan, H.; and Bansal, M. 2021. Unifying vision-and-language tasks via text generation. In *International Conference on Machine Learning*, 1931–1942. PMLR.
- Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70): 1–53.
- Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.
- Fan, L.; Li, T.; Qin, S.; Li, Y.; Sun, C.; Rubinstein, M.; Sun, D.; He, K.; and Tian, Y. 2024. Fluid: Scaling autoregressive text-to-image generative models with continuous tokens. *arXiv preprint arXiv:2410.13863*.
- Fu, C.; Chen, P.; Shen, Y.; Qin, Y.; Zhang, M.; Lin, X.; Yang, J.; Zheng, X.; Li, K.; Sun, X.; et al. 2023. MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models. *arXiv preprint arXiv:2306.13394*.
- Ganz, R.; Kittenplon, Y.; Aberdam, A.; Avraham, E. B.; Nuriel, O.; Mazor, S.; and Litman, R. 2024. Question Aware Vision Transformer for Multimodal Reasoning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13861–13871. IEEE Computer Society.
- Geigle, G.; Liu, C.; Pfeiffer, J.; and Gurevych, I. 2023. One does not fit all! On the Complementarity of Vision Encoders for Vision and Language Tasks. In Can, B.; Mozes, M.; Cahyawijaya, S.; Saphra, N.; Kassner, N.; Ravfogel, S.; Ravichander, A.; Zhao, C.; Augenstein, I.; Rogers, A.; Cho, K.; Grefenstette, E.; and Voita, L., eds., *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, 97–117. Toronto, Canada: Association for Computational Linguistics.
- Ghosh, A.; Acharya, A.; Saha, S.; Jain, V.; and Chadha, A. 2024. Exploring the Frontier of Vision-Language Models: A Survey of Current Methodologies and Future Directions. *CoRR*.
- Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Gurari, D.; Li, Q.; Stangl, A. J.; Guo, A.; Lin, C.; Grauman, K.; Luo, J.; and Bigham, J. P. 2018. VizWiz Grand Challenge: Answering Visual Questions from Blind People. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3608–3617. IEEE.
- Hadsell, R.; Chopra, S.; and LeCun, Y. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR’06)*, volume 2, 1735–1742. IEEE.
- He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. IEEE Computer Society.
- Hu, E. J.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.
- Kar, O. F.; Tonioni, A.; Poklukur, P.; Kulshrestha, A.; Zamir, A.; and Tombari, F. 2024. BRAVE: Broadening the visual encoding of vision-language models. In *European Conference on Computer Vision*, 113–132. Springer.
- Li, B.; Lin, Z.; Peng, W.; de Dieu Nyandwi, J.; Jiang, D.; Ma, Z.; Khanuja, S.; Krishna, R.; Neubig, G.; and Ramanan, D. 2024a. NaturalBench: Evaluating Vision-Language Models

- on Natural Adversarial Samples. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Li, B.; Wang, R.; Wang, G.; Ge, Y.; Ge, Y.; and Shan, Y. 2023a. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*.
- Li, J.; Kementchedjhieva, Y.; Fierro, C.; and Søgaard, A. 2024b. Do Vision and Language Models Share Concepts? A Vector Space Alignment Study. *Transactions of the Association for Computational Linguistics*, 12: 1232–1249.
- Li, J.; Selvaraju, R.; Gotmare, A.; Joty, S.; Xiong, C.; and Hoi, S. C. H. 2021. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34: 9694–9705.
- Li, Y.; Du, Y.; Zhou, K.; Wang, J.; Zhao, X.; and Wen, J.-R. 2023b. Evaluating Object Hallucination in Large Vision-Language Models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.
- Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.
- Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2024a. Improved Baselines with Visual Instruction Tuning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 26286–26296. IEEE.
- Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023a. Visual instruction tuning. *Advances in neural information processing systems*, 36: 34892–34916.
- Liu, K.; Fu, Z.; Chen, C.; Jin, S.; Chen, Z.; Tao, M.; Jiang, R.; and Ye, J. 2024b. Category-Extensible Out-of-Distribution Detection via Hierarchical Context Descriptions. In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Liu, Y.; Duan, H.; Zhang, Y.; Li, B.; Zhang, S.; Zhao, W.; Yuan, Y.; Wang, J.; He, C.; Liu, Z.; et al. 2024c. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, 216–233. Springer.
- Liu, Y.; Wang, K.; Shao, W.; Luo, P.; Qiao, Y.; Shou, M. Z.; Zhang, K.; and You, Y. 2023b. MLLMs-Augmented Visual-Language Representation Learning. *CoRR*.
- Lu, P.; Mishra, S.; Xia, T.; Qiu, L.; Chang, K.-W.; Zhu, S.-C.; Tafjord, O.; Clark, P.; and Kalyan, A. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35: 2507–2521.
- Masry, A.; Long, D. X.; Tan, J. Q.; Joty, S.; and Hoque, E. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*.
- Mathew, M.; Karatzas, D.; and Jawahar, C. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2200–2209.
- Mayilvahanan, P.; Wiedemer, T.; Rusak, E.; Bethge, M.; and Brendel, W. 2024a. Does CLIP’s generalization performance mainly stem from high train-test similarity? In *The Twelfth International Conference on Learning Representations*.
- Mayilvahanan, P.; Zimmermann, R. S.; Wiedemer, T.; Rusak, E.; Juhos, A.; Bethge, M.; and Brendel, W. 2024b. In Search of Forgotten Domain Generalization. In *ICML 2024 Workshop on Foundation Models in the Wild*.
- Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PmLR.
- Shahgir, H. S.; Sayeed, K. S.; Bhattacharjee, A.; Ahmad, W. U.; Dong, Y.; and Shahriyar, R. 2024. IllusionVQA: A Challenging Optical Illusion Dataset for Vision Language Models. In *First Conference on Language Modeling*.
- Shi, M.; Liu, F.; Wang, S.; Liao, S.; Radhakrishnan, S.; Huang, D.-A.; Yin, H.; Sapra, K.; Yacoob, Y.; Shi, H.; et al. 2024. Eagle: Exploring The Design Space for Multimodal LLMs with Mixture of Encoders. *CoRR*.
- Singh, A.; Natarajan, V.; Shah, M.; Jiang, Y.; Chen, X.; Batra, D.; Parikh, D.; and Rohrbach, M. 2019. Towards VQA Models That Can Read. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8309–8318. IEEE.
- Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–9. IEEE.
- Team, C. 2024. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*.
- Tong, P.; Brown, E.; Wu, P.; Woo, S.; IYER, A. J. V.; Akula, S. C.; Yang, S.; Yang, J.; Middepogu, M.; Wang, Z.; et al. 2024. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37: 87310–87356.
- Vyskočil, J.; and Pícek, L. 2023. VinVL+L: Enriching Visual Representation with Location Context in VQA. In *Computer Vision Winter Workshop, CEUR Workshop Proceedings*. Krems an der Donau, Austria.
- Wang, P.; Bai, S.; Tan, S.; Wang, S.; Fan, Z.; Bai, J.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; et al. 2024. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.
- Wang, X.; Chen, Y.; and Zhu, W. 2021. A survey on curriculum learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(9): 4555–4576.
- Xu, H.; Xie, S.; Tan, X.; Huang, P.-Y.; Howes, R.; Sharma, V.; Li, S.-W.; Ghosh, G.; Zettlemoyer, L.; and Feichtenhofer, C. 2024. Demystifying CLIP Data. In *The Twelfth International Conference on Learning Representations*.
- Yao, Y.; Yu, T.; Zhang, A.; Wang, C.; Cui, J.; Zhu, H.; Cai, T.; Li, H.; Zhao, W.; He, Z.; et al. 2024. MiniCPM-V: A GPT-4V Level MLLM on Your Phone. *CoRR*.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36: 46595–46623.

Zhu, D.; Chen, J.; Haydarov, K.; Shen, X.; Zhang, W.; and Elhoseiny, M. 2024. ChatGPT Asks, BLIP-2 Answers: Automatic Questioning Towards Enriched Visual Descriptions. *Transactions on Machine Learning Research*.

Reproducibility Checklist

Instructions for Authors:

This document outlines key aspects for assessing reproducibility. Please provide your input by editing this .tex file directly.

For each question (that applies), replace the “Type your response here” text with your answer.

Example: If a question appears as

```
\question{Proofs of all novel claims  
are included} {(yes/partial/no)}  
Type your response here
```

you would change it to:

```
\question{Proofs of all novel claims  
are included} {(yes/partial/no)}  
yes
```

Please make sure to:

- Replace **ONLY** the “Type your response here” text and nothing else.
- Use one of the options listed for that question (e.g., **yes**, **no**, **partial**, or **NA**).
- **Not** modify any other part of the `\question` command or any other lines in this document.

You can `\input` this .tex file right before `\end{document}` of your main file or compile it as a stand-alone document. Check the instructions on your conference’s website to see if you will be asked to provide this checklist with your paper or separately.

1. General Paper Structure

- 1.1. Includes a conceptual outline and/or pseudocode description of AI methods introduced (yes/partial/no/NA) **yes**
- 1.2. Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (yes/no) **yes**
- 1.3. Provides well-marked pedagogical references for less-familiar readers to gain background necessary to replicate the paper (yes/no) **yes**

2. Theoretical Contributions

- 2.1. Does this paper make theoretical contributions? (yes/no) **no**

If yes, please address the following points:

- 2.2. All assumptions and restrictions are stated clearly and formally (yes/partial/no) **NA**
- 2.3. All novel claims are stated formally (e.g., in theorem statements) (yes/partial/no) **NA**
- 2.4. Proofs of all novel claims are included (yes/partial/no) **NA**
- 2.5. Proof sketches or intuitions are given for complex and/or novel results (yes/partial/no) **NA**
- 2.6. Appropriate citations to theoretical tools used are given (yes/partial/no) **NA**
- 2.7. All theoretical claims are demonstrated empirically to hold (yes/partial/no/NA) **NA**
- 2.8. All experimental code used to eliminate or disprove claims is included (yes/no/NA) **NA**

3. Dataset Usage

- 3.1. Does this paper rely on one or more datasets? (yes/no) **yes**

If yes, please address the following points:

- 3.2. A motivation is given for why the experiments are conducted on the selected datasets (yes/partial/no/NA) **yes**
- 3.3. All novel datasets introduced in this paper are included in a data appendix (yes/partial/no/NA) **yes**
- 3.4. All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no/NA) **yes**
- 3.5. All datasets drawn from the existing literature (potentially including authors’ own previously published work) are accompanied by appropriate citations (yes/no/NA) **yes**
- 3.6. All datasets drawn from the existing literature (potentially including authors’ own previously published work) are publicly available (yes/partial/no/NA) **yes**
- 3.7. All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisfying (yes/partial/no/NA) **NA**

4. Computational Experiments

4.1. Does this paper include computational experiments?
(yes/no) [yes](#)

If yes, please address the following points:

4.2. This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting (yes/partial/no/NA) [yes](#)

4.3. Any code required for pre-processing data is included in the appendix (yes/partial/no) [no](#)

4.4. All source code required for conducting and analyzing the experiments is included in a code appendix (yes/partial/no) [no](#)

4.5. All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no) [yes](#)

4.6. All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from (yes/partial/no) [yes](#)

4.7. If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results (yes/partial/no/NA) [NA](#)

4.8. This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks (yes/partial/no) [yes](#)

4.9. This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics (yes/partial/no) [yes](#)

4.10. This paper states the number of algorithm runs used to compute each reported result (yes/no) [NA](#)

4.11. Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information (yes/no) [yes](#)

4.12. The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank) (yes/partial/no) [no](#)

4.13. This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments (yes/partial/no/NA) [yes](#)