

Instruction Tuning with and without Context: Behavioral Shifts and Downstream Impact

Hyunji Lee^{1*} Seunghyun Yoon² Yunjae Won³ Hanseok Oh⁴
Geewook Kim^{3,5} Trung Bui² Franck Dernoncourt² Elias Stengel-Eskin⁶
Mohit Bansal¹ Minjoon Seo³

¹ UNC Chapel Hill ² Adobe Research ³ KAIST AI
⁴ Mila – Quebec AI Institute ⁵ NAVER Cloud AI ⁶ The University of Texas at Austin

Abstract

Instruction tuning is a widely used approach to improve the instruction-following ability of large language models (LLMs). Instruction-tuning datasets typically include a mixture of context-augmented and context-free examples, yet prior work has largely combined these data types without examining their distinct effects. In this paper, we investigate how training LLMs with or without context affects model behavior and downstream performance. First, in the text domain, we show that LLMs trained with context attend more strongly to the provided knowledge, achieving better grounding. We also observe that context-augmented training shifts how LLMs use knowledge: models store and leverage less on parametric knowledge and instead depend more on the provided context. Second, we observe that using LLM trained with context-augmented data as the backbone for vision-language models reduces hallucination and improves grounding in the visual domain. Finally, we explore practical strategies for real-world deployments where context availability varies. We show that maintaining separate context-augmented and context-free models and routing inputs between them yields more robust overall performance than training a single mixed model, as it better preserves their complementary strengths¹.

1 Introduction

Large language models (LLMs) are often adapted to follow user instructions through instruction tuning, which finetunes the model on pairs of instructions and responses so that the models learn to operate in desired ways (Wei et al., 2021; Ouyang et al., 2022; Sanh et al., 2021). Instruction-tuning datasets (Taori et al., 2023; Conover et al., 2023) vary in whether each example is augmented

with external context (e.g., supporting documents) or is presented without any context. In practice, researchers often combine both data types to expose models to a broad range of instructions: context-augmented examples can help disambiguate prompts, support tasks such as summarization, and teach models to use provided evidence, while context-free examples encourage open-ended reasoning and general instruction following. Yet most prior work simply *mixes* these two data sources during training without examining their individual effects. This leaves an open question: *how does training with context-augmented or context-free instruction tuning data shape model knowledge and behaviors, and how do these differences transfer to other applications such as vision-language models?*

In this paper, we analyze these questions by comparing two LLM variants, CTX-LLM, which is trained on context-augmented instruction data, and NOCTX-LLM, which is trained on context-free instruction data. Specifically, we address three research questions: **(RQ1)** How do CTX-LLM and NOCTX-LLM differ in performance and knowledge use? **(RQ2)** How does using CTX-LLM and NOCTX-LLM as backbones for vision-language adaptation affect performance on vision-language tasks? **(RQ3)** How can these insights guide when to use CTX-LLM and NOCTX-LLM, and how to combine them effectively for downstream applications?

We observe that CTX-LLM achieves higher performance on information-seeking tasks when context is provided compared to NOCTX-LLM, as it implicitly learns to attend more strongly to the given evidence. Also, CTX-LLM maintains strong general language understanding, demonstrating solid language comprehension and reasoning ability. However, CTX-LLM performance drops on knowledge-intensive tasks without context, where the model must rely on its own para-

* Most of the work was done at KAIST.

¹<https://github.com/kaistAI/Effects-of-Context-in-Instruction-Tuning>

metric knowledge. Our analysis suggests that this degradation arises because training with context shifts the model’s reliance away from internal parametric memory and toward the provided context.

Next, we analyze how differences in LLM behavior, driven by the characteristics of the instruction-tuning data, affect performance on vision-language tasks. We compare models that use either CTX-LLM or NOCTX-LLM as the *backbone* for vision-language adaptation, applying the same vision-language alignment procedure in both cases. We find that using CTX-LLM as the backbone (CTX-VLM) reduces hallucination and improves grounding in the input image, showing that the grounding ability learned in the text domain successfully transfers to the visual domain as well. Moreover, CTX-VLM maintains stable factual accuracy across long generated responses, including facts expressed later in the sequences, where models often degrade and hallucinate (Lee et al., 2024b). Finally, CTX-VLM preserves strong performance on general vision-language understanding and reasoning benchmarks.

Finally, we investigate how these insights can inform the design and adaptation of instruction-tuning datasets for different downstream applications. Our earlier analysis shows that CTX-LLM is better suited for tasks that use provided context as the knowledge source, while NOCTX-LLM excels when tasks must rely on the model’s internal knowledge. However, many real-world applications require *both* capabilities. We therefore examine two approaches. First, we study the common practice of *mixing* context-augmented and context-free data into a single training set. Varying their ratio with a fixed total size shows that increasing context-augmented data improves performance on context-based tasks but slightly degrades parametric knowledge use, with roughly a 50/50 mix giving the most balanced single-model performance. Second, we investigate a *routing* setup: we train CTX-LLM and NOCTX-LLM separately and route inputs to the appropriate model using a simple heuristic of whether external context is provided. The routing setup consistently outperforms the mixed model, suggesting that keeping the two models separate and routing inputs is a practical way to preserve both context-based and parametric knowledge use.

2 Experimental Setup

In Section 2.1, we compare models trained on instruction tuning dataset with and without context. The following sections describe the experimental setup used to train and evaluate these two LLMs in both the text-only domain (Section 2.2) and the vision-language domain (Section 2.3). See Appendix A for more details.

2.1 Comparison Models: CTX-LLM vs. NOCTX-LLM

Instruction tuning a model without context (NOCTX-LLM) involves training it to generate responses r given an instruction i by minimizing the negative log-likelihood of the response tokens:

$$L(\theta) \approx -\mathbb{E}_{(i,r)} \sum_{t_k \in r} \log P_{\theta}(t_k | i, t_{<k}),$$

where $P_{\theta}(t_k | i, t_{<k})$ is the probability assigned by an autoregressive language model with parameters θ to the next token t_k , conditioned on the instruction i and the preceding tokens $t_{<k}$.

Instruction tuning a model with context (CTX-LLM) follows the same objective but includes an additional input c , representing external knowledge. During training, the context is provided alongside the instruction and prepended to the target response:

$$L(\theta) \approx -\mathbb{E}_{(i,c,r)} \sum_{t_k \in r} \log P_{\theta}(t_k | i, c, t_{<k}).$$

Following the design of previous work (Asai et al., 2024; Lee et al., 2024a) and our dataset construction, when training CTX-LLM, the relevant context c is prepended to the corresponding sentences in the target response r . The loss is computed only on the response tokens, excluding context itself. We provide additional experiments validating this design choice in Appendix A.

At inference time, both models follow the same generation procedure. If external knowledge is available, it is provided as context; otherwise, the model generates a response from the instruction alone.

2.2 Text Domain

Datasets & Evaluation Metrics For training, we use the 29k dataset from Self-RAG (Asai et al., 2024), constructed by augmenting instruction tuning datasets with relevant context identified at the

sentence level and incorporated when available². For evaluation, we experiment over 11 information-seeking datasets, including NQ (Kwiatkowski et al., 2019), TriviaQA (TQA) (Joshi et al., 2017), zsRE (Levy et al., 2017), T-rex (Elsahar et al., 2018), and HotpotQA (HQA) (Yang et al., 2018), using the versions provided in KILT (Petroni et al., 2021). For all experiments, we use the context provided in the original dataset. We also include DROP (Dua et al., 2019), SQuAD (Rajpurkar et al., 2016), SWDE (Lockard et al., 2019), and FDA (Arora et al., 2023), for which we use the version curated by Based (Arora et al., 2024). Additionally, we evaluate on two benchmarks specifically designed to highlight grounding failures in language models: NQ Conflict (NQ-C) (Zhou et al., 2023) and the dataset from CORG (Lee et al., 2025). For the dataset from CORG, we follow (Lee et al., 2025) in reporting the D-F1 metric, and for the rest, we evaluate using answer accuracy. We evaluate over 7 downstream tasks for general LLM performance: PIQA (Bisk et al., 2020), Social IQa (Sap et al., 2019), Winogrande (Sakaguchi et al., 2019), HellaSwag (Zellers et al., 2019), LAMBADA-OpenAI (Paperno et al., 2016), ARC-Challenge, and ARC-Easy (Clark et al., 2018)) using lm-evaluation-harness (Gao et al., 2024).

Training Details We conduct experiments on three pretrained models: Llama 2 7B (Touvron et al., 2023), Llama 3.1 8B (Grattafiori et al., 2024), and Qwen 2.5 7B (Yang et al., 2024)³. We train the full model for three epochs with a batch size of 128, a learning rate of 2e-5, and the AdamW optimizer. All training are conducted using 4 NVIDIA A100 80G GPUs.

2.3 Vision-Language Domain

Datasets & Evaluation Metrics When training the vision-language alignment, we use the training dataset from LLaVA (Liu et al., 2023b) for both the pretraining and finetuning stages. We evaluate over four hallucination benchmarks AMBER (Wang et al., 2023), POPE (Li et al., 2023b), ImageInWords (Garg et al., 2024), and LLaVA-Wild (Liu et al., 2023b) to measure its grounding ability to provided image. For POPE, we report the average F1 score across all splits. For ImageInWords, we adopt the evaluation from CapMAS (Lee et al.,

²See Appendix A for details on filtering over Self-RAG training datasets.

³If not specified, we use Llama 3.1 8B as the base model for all analyses, with full-parameter fine-tuning.

2024b), a GPT-based approach for fine-grained factuality assessment. For LLaVA-Wild, to better target hallucination detection, we modify the rubric to explicitly penalize hallucinations. We evaluate over four downstream tasks for general VLM performance: MMBench (Liu et al., 2023d), ScienceQA (Lu et al., 2022), MME (Sun et al., 2023a), and GQA (Hudson and Manning, 2019).

Training Details For all models, we follow a widely used two-stage process (Chen et al., 2024; Liu et al., 2023b): first, a pretraining stage for feature alignment, and second, an end-to-end finetuning stage. Specifically, we follow the training configurations introduced in LLaVA (Liu et al., 2023b), and conduct all experiments on 8 NVIDIA A100 80G GPUs. Note that the vision-language alignment procedure is identical across all experiments, and only the backbone for the LLM varies.

3 RQ1: How do CTX-LLM and NOCTX-LLM differ in performance and knowledge use?

In this section, we investigate how training an LLM on instruction-tuning datasets with context (CTX-LLM) versus without context (NOCTX-LLM) affects its behavior on information-seeking tasks and general language understanding.

3.1 CTX-LLM Improves Grounding

Table 1 shows that CTX-LLM, LLMs trained with context, consistently achieve higher performance on information-seeking datasets when provided with relevant context at inference time. Across a range of pretrained backbones (Llama 2 7B, Llama 3.1 8B, and Qwen 2.5 7B), CTX-LLM yields an average absolute improvement of 5.5% over NOCTX-LLM, which are trained without context. The gains are especially pronounced on benchmarks such as NQ-C and CORG, which require handling counterfactual or complex knowledge in the provided context; CTX-LLM achieves an average of 8.6% absolute improvement compared to an average of 4.8% improvement on the remaining datasets. These results suggest that training models on instruction-tuning datasets containing context strengthens their grounding ability, even in cases where standard models often misinterpret or generate incorrect responses.

BaseModel	ModelType	NQ	TQA	zsRE	T-rex	HQA	NQ-C	Corg	Drop	Squad	SWDE	FDA	Avg
Llama2	NoCTX-LLM	42.3	69.0	51.2	69.8	45.5	54.1	17.3	33.8	42.9	82.3	73.4	52.9
	CTX-LLM	55.8	72.1	65.1	60.2	49.0	75.1	19.1	38.7	58.3	81.0	76.3	59.2
Llama3.1	NoCTX-LLM	48.0	70.2	57.5	62.7	48.0	60.7	18.0	39.0	64.9	92.0	74.0	57.7
	CTX-LLM	46.2	72.6	62.6	63.4	50.6	72.3	19.4	44.0	69.5	95.0	80.9	61.5
Qwen2.5	NoCTX-LLM	57.2	73.5	67.0	68.9	60.0	58.2	17.9	30.9	62.7	84.1	81.9	60.2
	CTX-LLM	66.1	84.6	72.8	74.6	54.2	71.0	20.7	49.5	60.5	88.2	89.4	66.5

Table 1: Performance comparison of instruction-tuned models trained with context (CTX-LLM) vs. without context (NOCTX-LLM) across 11 information-seeking datasets, using three base models: Llama2 7B, Llama3.1 8B, and Qwen2.5 7B.

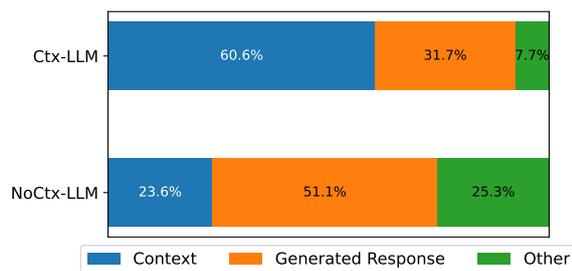


Figure 1: Avg. rate at which the model attends most to each input segment: context, generated response, or other (e.g., system prompts) during generation.

3.2 Context-Augmented Training Shifts Usage of Provided External Knowledge

Attention Weight We observe that while training with context does not *explicitly* guide the model to attend more strongly to the external context, the inclusion of relevant context in training instances implicitly encourages this behavior and enhances grounding; CTX-LLM shows better grounding ability than NOCTX-LLM. Figure 1 shows the average ratio of which part of the input (Context, Generated Response, or Other) the models attend to the most when generating responses for instances in NQ. For each generated token, we identify the input segment that receives the highest attention weight, compute the proportion of tokens attending to each segment per instance, and then average those proportions across all instances.

Results show that CTX-LLM assigns greater attention to the given context (*blue*), whereas NOCTX-LLM tends to focus more on its own previously generated responses (*orange*). A similar pattern appears in the full attention map provided in Appendix B.2. These results suggest that training a model with context included induces a mechanistic shift in the model’s generation process, leading the model to assign greater attention to relevant context tokens rather than to self-generated content,

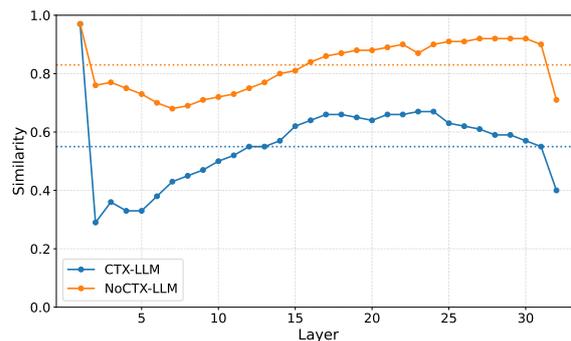


Figure 2: Layer-wise cosine similarity of hidden states during inference with and without external context for NOCTX-LLM and CTX-LLM; lower similarity indicates stronger reliance on external knowledge. Dotted horizontal lines denote the average similarity for each model.

thereby producing more grounded outputs.

Layer-wise Activation We additionally conducted a layer-wise analysis to examine how the knowledge utilization pattern differs between CTX-LLM and NOCTX-LLM. We calculate the cosine similarity between hidden states generated during inference with or without context on the NQ dataset using models trained with Llama-3.1 as a base model. High similarity implies the model ignores the context in favor of internal memory, whereas low similarity indicates active utilization.

Result in Figure 2 shows that NOCTX-LLM exhibits high similarity across layers (avg. 0.83), suggesting minimal reliance on external context. In contrast, CTX-LLM shows a much lower average (0.55). It especially shows a sharp drop to 0.3 in the initial layers, suggesting that the model integrates external knowledge early, effectively altering representations in all subsequent layers.

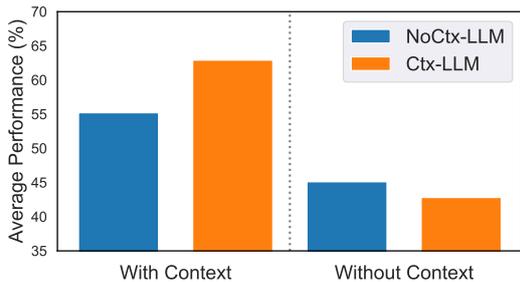


Figure 3: Average Performance of CTX-LLM and NOCTX-LLM across inference setups. The x-axis indicates whether context is provided at inference: *With Context* uses external context, while *Without Context* requires the model to rely on its own parametric knowledge.

3.3 Context-Augmented Training Alters Reliance on Parametric Knowledge

When evaluating on datasets where no relevant context is provided, which forces models to rely solely on their parametric knowledge, we observe that CTX-LLM performs less effectively than NOCTX-LLM (Figure 3). We hypothesize that training with context-augmented data shifts how models use knowledge: NOCTX-LLM tends to encode more information directly in its parameters, while CTX-LLM learns to depend on provided context. This aligns with our earlier observation that CTX-LLM attends more strongly to external evidence.

To test this hypothesis, we manipulate the relationship between the knowledge contained in the training data and the model’s parametric knowledge. In addition to the *original (Ori)* dataset, where the provided context and gold answer align with the model’s prior knowledge, we construct a *counterfactual (CF)* dataset, where the context and answer contradict the model’s parametric knowledge. We then train four models by crossing context availability (CTX vs. NOCTX) with knowledge alignment (Ori vs. CF), and evaluate each model under four inference settings that vary (i) whether external context is available (+Ctx / -Ctx) and (ii) whether the context and answer are original or counterfactual (Ori / CF).

Results in Figure 4) suggest that training on context-augmented (CTX) versus context-free data (NOCTX) leads to *differences in how models internalize and use knowledge*. NOCTX models tend to memorize training knowledge more strongly, achieving the highest score in inference setting where they must rely on their parametric knowl-

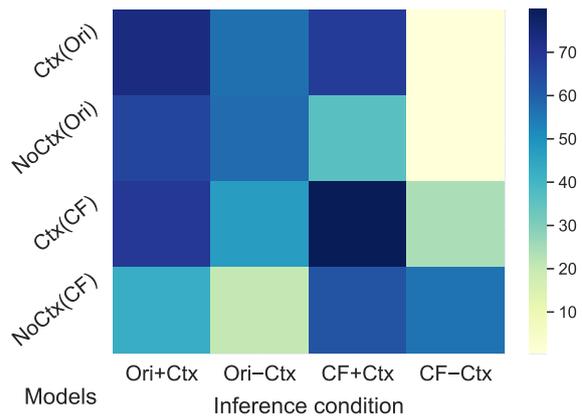


Figure 4: Accuracy by models trained on different datasets (*Models*) and evaluated under different inference conditions (*Inference condition*). *Original (Ori)* refers to knowledge aligned with the model’s parametric knowledge, while *Counterfactual (CF)* denotes counterfactual knowledge. “+” or Ctx indicates that context is provided; “-” or NoCtx indicates no context is provided during training or inference.

edge of that same knowledge. Specifically, when trained on counterfactual knowledge, NOCTX models show greater forgetting of prior knowledge and stronger memorization of the updated training signal, whereas CTX models retain more of their original knowledge but memorize counterfactual knowledge less strongly. In addition, CTX models consistently achieve higher performance whenever context is available, regardless of whether the context supports original or counterfactual answers.

These findings suggest that context-augmented training shifts the model’s reliance from parametric memory toward external context. Models trained with context become better at using retrieved evidence but may encode less new knowledge directly in their parameters compared to models trained without context. Details of counterfactual dataset construction and results are in Appendix B.3.

3.4 CTX-LLM Show High Performance in General Language Understanding

To compare model behavior or performance of CTX-LLM and NOCTX-LLM over other tasks apart from information-seeking tasks, we evaluate them on seven widely used downstream benchmarks: PIQA, Social IQa, Winogrande, HellaSwag, LAMBADA-OpenAI, ARC-Challenge, and ARC-Easy. Results in Table 3 show that CTX-LLM tend to show higher or comparable performance compared to NOCTX-LLM, achieving an average score of 68.4 compared to 68.2. This suggests that train-

BaseModel	ModelType	Pope	Amber						Llava-W	Caption
		F1	CHAIR (\downarrow)	Cover (\uparrow)	Hal (\uparrow)	Cog (\downarrow)	F1 (\uparrow)	AMBER (\uparrow)	GPT4-Eval	F1
Llama2	NoCTX-VLM	84.7	9.3	47.8	38.8	5.0	65.0	77.9	53.4	54.1
	CTX-VLM	85.5	7.3	48.0	30.5	3.4	71.4	82.1	70.9	55.9
Llama3.1	NoCTX-VLM	87.3	9.1	53.8	39.8	5.9	66.5	78.7	55.7	54.5
	CTX-VLM	87.7	8.6	54.6	47.2	4.9	71.7	81.6	74.2	56.8
Qwen2.5	NoCTX-VLM	87.9	8.5	49.1	34.4	4.8	70.8	81.2	60.1	56.7
	CTX-VLM	88.8	7.2	52.9	40.2	5.1	72.3	82.6	78.5	58.7

Table 2: Performance of VLM using NOCTX-LLM as the LLM backbone (NOCTX-VLM) and CTX-LLM as the LLM backbone (CTX-VLM) across four hallucination benchmarks, using three base models (Llama2 7B, Llama3.1 8B, Qwen 2.5 7B). The first row indicates the evaluation dataset, and the second row shows the metric.

Ctx	PI	SI	WI	HS	LA	AC	AE	Avg
F	81.9	48.6	72.6	80.6	75.6	56.7	82.2	68.2
T	82.9	49.0	73.4	80.3	76.3	56.1	83.0	68.4

Table 3: Performance of CTX-LLM (Ctx=T) and NOCTX-LLM (Ctx=F) using Llama3.1 8B as base model, across seven widely used downstream benchmarks. PI is PiQA, SI is SocialIQA, WI is Winogrande, HS is Hellaswag, LA is LAMBADA-OpenAI, AC is ARC-challenge, and AE is ARC-Easy.

ing with context-augmented data tends to preserve or even enhance general language understanding across diverse tasks.

4 RQ2: How does using CTX-LLM or NOCTX-LLM as the backbone for vision-language adaptation influence performance on vision-language tasks?

In this section, we analyze how training an LLM with or without context (CTX-LLM vs. NOCTX-LLM) affects its performance when used as the backbone for vision-language adaptation. We compare two configurations: CTX-VLM, which uses an LLM trained with context (CTX-LLM), and NOCTX-VLM, which uses one trained without context (NOCTX-LLM), while keeping all other training and alignment procedures identical.

4.1 Using CTX-LLM as a backbone improves grounding in vision-language models

To assess grounding in vision-language models, we evaluate CTX-VLM and NOCTX-VLM on four hallucination benchmarks: POPE, AMBER, LLaVA-Wild, and ImageInWords. Table 2 shows that CTX-VLM consistently outperforms NOCTX-VLM, achieving higher accuracy and reduced hallucination. This suggests that CTX-VLM generates responses that are more faithfully grounded in

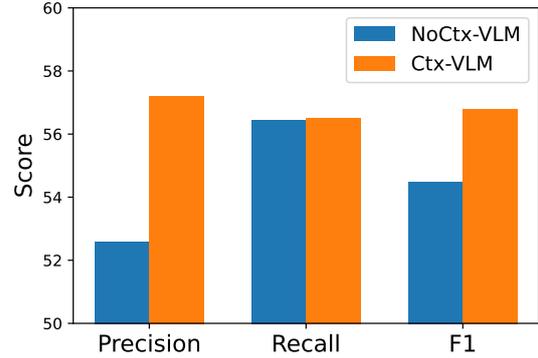


Figure 5: Precision, Recall, and F1 score on the ImageInWords fine-grained captioning task, evaluated with CapMAS, comparing CTX-VLM and NOCTX-VLM using Llama3.1 8B as base model.

the visual input rather than relying primarily on its parametric knowledge.

Notably, in LLaVA-Wild, the performance gap widens when the evaluation rubric penalizes hallucination more heavily: CTX-VLM surpasses NOCTX-VLM by +18.1% under the stricter rubric, compared to +8.8% under the original rubric. Similarly, as shown in Figure 5, CTX-VLM achieves stronger results on ImageInWords, with a notable gain in precision. These gains indicate that captions generated by CTX-VLM are more accurate and better grounded in the provided visual input. Together, these results highlight the advantage of using a context-augmented backbone (CTX-LLM) for reducing hallucination and improving grounding in vision-language models.

We hypothesize that this improved grounding ability stems from the generalization of grounding behaviors learned during instruction tuning with context-augmented data. Specifically, in CTX-LLM, models learn to effectively leverage provided external knowledge, and this grounding be-

Model	MMBench	ScienceQA	MME	GQA
NOCTX-VLM	68.6	78.4	1526.1	63.4
CTX-VLM	70.2	79.1	1534.4	64.1

Table 4: Comparison of NOCTX-VLM and CTX-VLM over four vision-language downstream tasks using Llama 3.1-8B as base model.

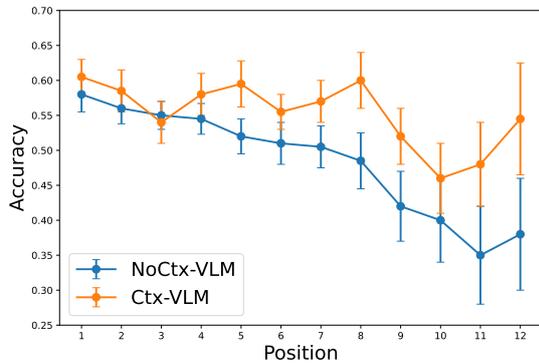


Figure 6: Avg. accuracy (y-axis) of atomic facts from generated responses as a function of their position (x-axis). Error bars indicate variance.

havior appears to *transfer when the external input shifts from textual context to visual information* in the vision-language setting (CTX-VLM).

4.2 CTX-VLM show robust performance across varying response lengths

Figure 6 presents fine-grained captioning performance on the ImageInWords benchmark, showing accuracy as a function of the position of knowledge within the generated response, evaluated using the CapMAS method (Lee et al., 2024b), which decomposes generated sentences into atomic fact units using GPT-4o and assesses their truthfulness based on the corresponding image and reference caption. The x-axis in the figure indicates the position at which a fact appears in the generated caption.

CTX-VLM maintains more stable accuracy across different positions, especially outperforming NOCTX-VLM at later positions, where the performance gap becomes increasingly pronounced. This suggests that CTX-VLM are more robust at preserving factual consistency throughout the entire generated response. A qualitative example of fine-grained caption in Appendix C.1 also illustrates this behavior; while the initial parts of the responses from NOCTX-VLM and CTX-VLM tend to be similar, their outputs diverge notably toward the end where NOCTX-VLM generates an incorrect response and CTX-VLM provides a more detailed

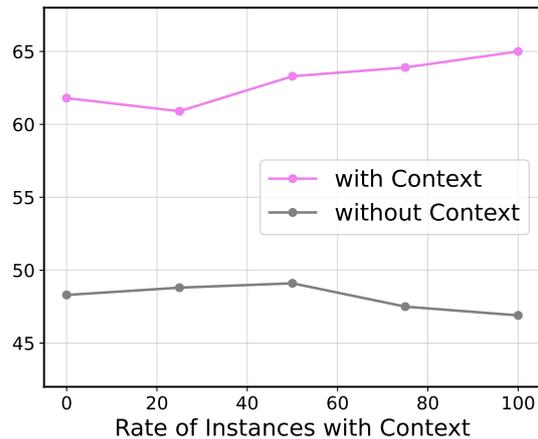


Figure 7: Performance of models (y-axis) trained with different proportions of context-augmented training instances (x-axis). The pink line shows performance when inference is performed with relevant context, while the gray line shows performance when inference is performed without context.

observation.

4.3 CTX-VLM shows robust performance on general VLM downstream tasks

To evaluate the performance of CTX-VLM and NOCTX-VLM beyond hallucination benchmarks, we test both models on four representative vision-language benchmarks: MMBench, ScienceQA, MME, and GQA. Table 4 shows that CTX-VLM achieves performance comparable to or higher than NOCTX-VLM, indicating that its improved grounding ability and reduced hallucination do not come at the expense of general vision-language understanding and reasoning.

5 RQ3: How can these insights guide when to use each model and combine them effectively for downstream applications?

Our earlier experiments show that CTX-LLM demonstrates strong grounding ability⁴ while maintaining general language understanding, making it a suitable choice for applications where relevant external context is available at inference time (e.g., vision-language tasks, retrieval-augmented LLMs). Conversely, NOCTX-LLM achieve higher performance when they must rely on their own

⁴In Appendix D.1, we further observe that using CTX-LLM with inference-time grounding methods yields higher performance than using NOCTX-LLM, reinforcing that CTX-LLM is better suited for grounding-based applications.

parametric knowledge without access to external evidence. However, many real-world scenarios require a model to effectively use *both* provided context and its internal knowledge.

In this section, we examine two approaches for combining the strengths of CTX-LLM and NOCTX-LLM in such applications. The first is *mixture training*, a common practice in instruction tuning, where a single model is trained on a mix of context-augmented and context-free examples. The second is *routing-based inference*, where two models, CTX-LLM and NOCTX-LLM, are trained separately, and inputs are routed to the appropriate model using a simple heuristic of whether relevant context is available.

Training with Mixture of Context-augmented and Context-free Data

We investigate how the proportion of context-augmented versus context-free training examples affects a model’s ability to ground in external context while retaining parametric knowledge. We keep the total number of training examples fixed and vary the proportion that is context-augmented: 0%, 25%, 50%, 75%, and 100%. As shown in Figure 7, increasing the rate of context-augmented training data generally improves performance when context is available at inference, while causing slight degradation when no context is provided. Notably, a 50% mix achieves the most balanced performance, maintaining strong grounding ability when context is available while preserving performance on knowledge-intensive tasks without context.

Routing Inputs Based on Context Availability

We experiment with a routing setup where inputs are directed to either CTX-LLM or NOCTX-LLM depending on whether relevant context is provided in inference step. Results in Figure 17 show that routing (purple line) performs well in both when inference with and without context. This suggests that the two models could be used in complementary ways; for example, when designing or extending mixture-of-experts architectures, one could include CTX-LLM and NOCTX-LLM as separate experts to leverage both grounding ability and parametric knowledge. We observe that this approach (56.7) consistently outperforms any single mixed model (55.6), suggesting that in practical deployments it is often more effective to *maintain separate experts and route between them* rather than rely on a single model trained on a mixed dataset. Performance scores are averaged over nine evalua-

tion datasets⁵ covering both context-available and context-free inference settings.

We also tested a LoRA-based approach, where everything is equal except that it is trained with LoRA rather than full parameters to make it lightweight. We observe similar trend with when training full parameters, suggesting a practical alternative to fully maintaining two models. More details regarding results are in Appendix D.2. Future work could extend to using a trainable router to decide dynamically whether context would improve performance, rather than relying on the simple heuristic of context presence.

6 Related Works

Instruction Tuning and Its Impact on Context Awareness and Knowledge Use

Several works have shown that instruction tuning influences on how LLMs use their parametric knowledge and given input context. Recent studies show loss of context awareness after instruction tuning⁶. Goyal et al. (2024) find that models tend to be less reliant on provided context under knowledge conflict as instruction tuning progresses. Similarly, Wang et al. (2024b) attributes this loss to role biases introduced by chat-style prompting templates. Other analysis explore how instruction tuning reshapes behavioral and representational properties of LLMs in attention distribution or attribution (Wu et al., 2023; Gao et al., 2023). Another line of work aim to enhance the model’s external context utilization through instruction tuning: overcoming the lost-in-the-middle problem (Liu et al., 2023c) in long-context inputs (An et al., 2024; Begin et al., 2025; He et al., 2023) or better grounding on given context (Lee et al., 2024a; Asai et al., 2024; Tian et al., 2023; Luo et al., 2023). Together, these studies examine how instruction tuning affects the utilization of parametric knowledge versus user-provided context. Our work extends this understanding by examining new axes of how training on datasets with or without relevant context affects a model’s grounding ability and its use of parametric knowledge.

Improving Grounding in Language and Vision-Language Models

Prior work have explored multiple directions to strengthen grounding.

⁵We exclude NQ-C and Corg from the 11 information-seeking datasets, as counterfactual answers are not reliably answerable without context.

⁶Please note that the training dataset used here do not contain external knowledge (NOCTX-LLM setting)

Inference-time methods modify decoding or introduce post-hoc revision pipelines to better incorporate external knowledge (Shi et al., 2023; Wang et al., 2024a; Gao et al., 2022; Chern et al., 2023). Training-time approaches aim to align models to external evidence, for example using preference optimization for factuality (Tian et al., 2023) or integrating retrieval and self-critique signals as in Self-RAG (Asai et al., 2024).

Improving grounding ability is also crucial for vision-language models (VLMs) to ensure that generated responses are based on the image rather than relying on the language model to produce plausible outputs, but not based on the image. Prior work has explored several directions for enhancing grounding in VLMs, including adjusting decoding (Favero et al., 2024; Leng et al., 2023), curating high-quality datasets (Liu et al., 2023a; Li et al., 2023a), and developing training strategies to strengthen visual grounding (Sun et al., 2023c; Ouali et al., 2024).

7 Conclusion

In this paper, we studied the impact of training LLMs with context-augmented data (CTX-LLM) versus context-free data (NOCTX-LLM). We observed that training with context shifts how model uses knowledge: reducing reliance on its parametric knowledge and encouraging stronger use of the provided context. This behavior generalizes beyond the text domain to the visual domain, leading to improved performance on hallucination benchmarks. Moreover, our exploration on practical deployment strategies suggests that rather than mixing both data types into a single model, maintaining separate CTX-LLM and NOCTX-LLM and routing yields stronger overall performance.

Limitations

Our experiments are conducted on 7B scale models due to computational constraints. However, we tested over three different base models and observed a consistent trend, suggesting that our findings are likely to generalize to various, larger models. In addition, we conduct a similar trend for experiments at the 14B and 32B parameter scales using LoRA in Table 8. Additionally, our study assumes that the provided context is reliable and relevant. Future work could explore scenarios where the context may be unreliable, noisy, or partially irrelevant, requiring the model to assess the trustworthiness of external information before grounding on it.

Acknowledgments

This work was partly supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (RS-2024-00397966, Development of a Cybersecurity Specialized RAG-based sLLM Model for Suppressing Gen-AI Malfunctions and Construction of a Publicly Demonstration Platform, 10%; RS-2024-00398115, Research on the reliability and coherence of outcomes produced by Generative AI, 10%)

References

- Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. 2024. Make your llm fully utilize the context. *Advances in Neural Information Processing Systems*, 37:62160–62188.
- Simran Arora, Aman Timalsina, Aaryan Singhal, Sabri Eyuboglu, Xinyi Zhao, Ashish Rao, Atri Rudra, and Christopher Ré. 2024. Just read twice: closing the recall gap for recurrent language models.
- Simran Arora, Brandon Yang, Sabri Eyuboglu, Avani Narayan, Andrew Hojel, Immanuel Trummer, and Christopher Ré. 2023. [Language models enable simple systems for generating structured views of heterogeneous data lakes](#). *Preprint*, arXiv:2304.09433.
- Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. [Self-RAG: Learning to retrieve, generate, and critique through self-reflection](#). In *The Twelfth International Conference on Learning Representations*.
- James Begin, Namit Agrawal, Eshan Singh, Yicheng Fu, Sean O’Brien, Vasu Sharma, and Kevin Zhu. 2025. [Pause-tuning for long-context comprehension: A lightweight approach to llm attention recalibration](#). *arXiv preprint arXiv:2502.20405*.
- Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. [Piqa: Reasoning about physical commonsense in natural language](#). In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.
- Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, and 1 others. 2024. [How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites](#). *Science China Information Sciences*, 67(12):220101.
- I Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, Pengfei Liu, and 1 others. 2023. [Factool: Factuality detection in generative ai—a tool augmented](#)

- framework for multi-task and multi-domain scenarios. *arXiv preprint arXiv:2307.13528*.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.
- Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. [Free dolly: Introducing the world’s first truly open instruction-tuned llm](#).
- Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *arXiv preprint arXiv:1903.00161*.
- Hady Elsahar, Pavlos Vougiouklis, Arslan Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- Alessandro Favero, Luca Zancato, Matthew Trager, Siddharth Choudhary, Pramuditha Perera, Alessandro Achille, Ashwin Swaminathan, and Stefan O Soatto. 2024. [Multi-modal hallucination control by visual information grounding](#). *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14303–14312.
- Changjiang Gao, Shujian Huang, Jixing Li, and Jiajun Chen. 2023. Roles of scaling and instruction tuning in language perception: Model vs. human attention. *arXiv preprint arXiv:2310.19084*.
- Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac’h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. [The language model evaluation harness](#).
- Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and 1 others. 2022. Rarr: Researching and revising what language models say, using language models. *arXiv preprint arXiv:2210.08726*.
- Roopal Garg, Andrea Burns, Burcu Karagol Ayan, Yonatan Bitton, Ceslee Montgomery, Yasumasa Onoe, Andrew Bunner, Ranjay Krishna, Jason Baldridge, and Radu Soricut. 2024. [Imagein-words: Unlocking hyper-detailed image descriptions](#). *Preprint*, arXiv:2405.02793.
- Sachin Goyal, Christina Baek, J Zico Kolter, and Aditi Raghunathan. 2024. Context-parametric inversion: Why instruction finetuning can worsen context reliance. *arXiv preprint arXiv:2410.10796*.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Junqing He, Kunhao Pan, Xiaoqun Dong, Zhuoyang Song, Yibo Liu, Qianguo Sun, Yuxin Liang, Hao Wang, Enming Zhang, and Jiaying Zhang. 2023. Never lost in the middle: Mastering long-context question answering with position-agnostic decompositional training. *arXiv preprint arXiv:2311.09198*.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.
- Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709.
- Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. [Unsupervised dense information retrieval with contrastive learning](#).
- Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. [triviaqa: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension](#). *arXiv e-prints*, arXiv:1705.03551.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*.
- Hyunji Lee, Franck Dernoncourt, Trung Bui, and Seunghyun Yoon. 2025. CORG: Generating answers from complex, interrelated contexts. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*.
- Hyunji Lee, Doyoung Kim, Jihoon Jun, SeJune Joo, Joel Jang, Kyoung-Woon On, and Minjoon Seo. 2024a. Semiparametric token-sequence co-supervision. *ACL*.
- Saehyung Lee, Seunghyun Yoon, Trung Bui, Jing Shi, and Sungroh Yoon. 2024b. Toward robust hyper-detailed image captioning: A multiagent approach and dual evaluation metrics for factuality and coverage. *arXiv preprint arXiv:2412.15484*.

- Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Li Bing. 2023. [Mitigating object hallucinations in large vision-language models through visual contrastive decoding](#). *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13872–13882.
- Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115*.
- Lei Li, Yuwei Yin, Shicheng Li, Liang Chen, Peiyi Wang, Shuhuai Ren, Mukai Li, Yazheng Yang, Jingjing Xu, Xu Sun, Lingpeng Kong, and Qi Liu. 2023a. [M3it: A large-scale dataset towards multi-modal multilingual instruction tuning](#). *ArXiv*, abs/2306.04387.
- Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023b. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*.
- Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. 2023a. Aligning large multi-modal model with robust instruction tuning. *arXiv preprint arXiv:2306.14565*.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning.
- Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023c. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.
- Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. 2023d. Mmbench: Is your multi-modal model an all-around player? *arXiv:2307.06281*.
- Colin Lockard, Prashant Shiralkar, and Xin Luna Dong. 2019. OpenCeres: When open information extraction meets the semi-structured web. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.
- Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*.
- Hongyin Luo, Yung-Sung Chuang, Yuan Gong, Tianhua Zhang, Yoon Kim, Xixin Wu, Danny Fox, Helen Meng, and James Glass. 2023. Sail: Search-augmented instruction learning. *arXiv preprint arXiv:2305.15225*.
- Yassine Ouali, Adrian Bulat, Brais Martinez, and Georgios Tzimiropoulos. 2024. Clip-dpo: Vision-language models as a source of preference for fixing hallucinations in lvlms. In *European Conference on Computer Vision*, pages 395–413. Springer.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. [The lambda dataset](#).
- Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *EMNLP*.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*.
- Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, and 1 others. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.
- Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473.
- Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Scott Wen-tau Yih. 2023. Trusting your evidence: Hallucinate less with context-aware decoding. *arXiv preprint arXiv:2305.14739*.

- Xinyu Sun, Peihao Chen, Liangwei Chen, Changhao Li, Thomas H Li, Mingkui Tan, and Chuang Gan. 2023a. Masked motion encoding for self-supervised video representation learning. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, Kurt Keutzer, and Trevor Darrell. 2023b. Aligning large multi-modal models with factually augmented rlhf.
- Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liangyan Gui, Yu-Xiong Wang, Yiming Yang, Kurt Keutzer, and Trevor Darrell. 2023c. [Aligning large multi-modal models with factually augmented rlhf](#). *ArXiv*, abs/2309.14525.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.
- Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D. Manning, and Chelsea Finn. 2023. [Fine-tuning language models for factuality](#). *ArXiv*, abs/2311.08401.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2024a. Adacad: Adaptively decoding to balance conflicts between contextual and parametric knowledge. *arXiv preprint arXiv:2409.07394*.
- Junyang Wang, Yuhang Wang, Guohai Xu, Jing Zhang, Yukai Gu, Haitao Jia, Ming Yan, Ji Zhang, and Jitao Sang. 2023. An llm-free multi-dimensional benchmark for mllms hallucination evaluation. *arXiv preprint arXiv:2311.07397*.
- Yihan Wang, Andrew Bai, Nanyun Peng, and Cho-Jui Hsieh. 2024b. On the loss of context-awareness in general instruction fine-tuning. *arXiv preprint arXiv:2411.02688*.
- Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.
- Xuansheng Wu, Wenlin Yao, Jianshu Chen, Xiaoman Pan, Xiaoyang Wang, Ninghao Liu, and Dong Yu. 2023. From language modeling to instruction following: Understanding the behavior shift in llms after instruction tuning. *arXiv preprint arXiv:2310.00492*.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *EMNLP*.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.
- Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2023. Context-faithful prompting for large language models. *arXiv preprint arXiv:2303.11315*.

A Experimental Setup

A.1 Comparison Models: CTX-LLM vs. NOCTX-LLM

Table 5 shows performance of various models trained on various design choices of provided context. There are three axes: whether the context exists in training dataset (Ctx Presence), whether the context is added to the input (instruction) or output (next to the corresponding response sentences) (Ctx Placement), or whether to calculate loss over the context or not (Ctx Loss). For case where context exists, we used the case where context is added to the output side with no loss over the context following previous works (Lee et al., 2024a; Asai et al., 2024). Also this showed highest performance. We also observed some interesting analysis by the choices when context is present (CTX-LLM):

Ctx Placement When comparing performance on model when training with context added to the input (instruction) or output (assistant response), we observe that overall performance is comparable across both settings, with a slight average improvement when context is added to the output, aligning with findings from prior work (Lee et al., 2024a; Asai et al., 2024), which suggest that placing context closer to the generation target improves grounding performance. Main difference for these two comes when there are multiple evidences for single example, thereby rather to concatenate all and add to user prompt or to separate each into relevant response sentence and place them in front.

Ctx Presence	Ctx Placement	Ctx Loss	NQ	TQA	zsRE	T-rex	HQA	NQ-C	Corg	Drop	Squad	SWDE	FDA	Avg
F	-	-	48.0	70.2	57.5	62.7	48.0	60.7	18.0	39.0	64.9	92.0	74.0	57.7
T	Input	T	42.9	69.0	60.2	57.9	43.5	33.6	11.2	18.0	54.5	90.8	71.2	50.3
T	Input	F	47.1	72.0	60.3	62.9	49.5	72.1	20.6	45.1	68.8	95.7	78.9	61.2
T	Output	T	45.8	71.7	60.0	64.1	48.1	47.0	13.1	30.3	63.8	94.3	76.9	55.9
T	Output	F	46.2	72.6	62.6	63.4	50.6	72.3	19.4	44.0	69.5	95.0	80.9	61.5

Table 5: Performance comparison over various context choices; Ctx Presence is whether the context is added to the response. Ctx Placement is whether the relevant context is added to the input (instruction) or output (before each corresponding response sentence). Ctx Loss is rather we calculate loss over the context or not. We train all models using Llama3.1 8B as base model and evaluate over 11 information-seeking datasets.

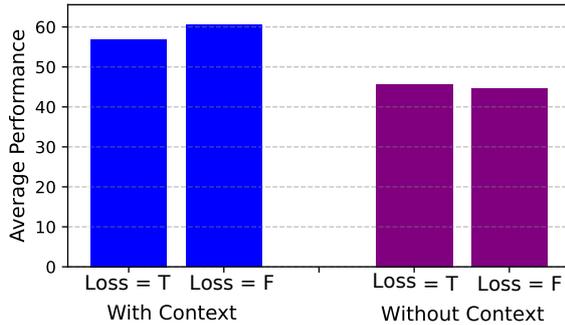


Figure 8: Performance of models trained with context, either with loss over the context (Loss = T) or without loss over the context (Loss = F). *With Context* indicates inference with additional context provided, while *Without Context* indicates inference without additional context.

Ctx Loss We experiment over two training settings: one where the loss is computed over both the context and the response (loss=T), and one where it is computed only over the response (loss=F). As shown in Figure 8, models trained without loss on the context (loss=F) perform better when context is available at inference. However, their performance drops to the same level or worse than model trained with loss on the context (loss=T) when no context is provided, thereby the model must rely on its parametric knowledge.

We hypothesize that this occurs because, under loss=T, the model must encode and store the given context in its parameters to minimize training loss, which later helps when no external context is available. In contrast, models trained with loss=F learn how to use provided context at inference rather than memorizing it, as also observed in Section 3.2.

A.2 Text Domain

Datasets For training, we use the 29k dataset from Self-RAG (Asai et al., 2024), which is constructed by augmenting instruction-tuning datasets with sentence-level relevant context, incorporated

when available. Following the filtering procedure from prior work (Lee et al., 2024a), we retain only instances with relevant context. Additionally, we exclude examples where the generation of counterfactual contexts fails, to facilitate more focused analysis and experimentation on counterfactual behavior (see Appendix B.3).

For evaluation, we experiment over 11 information-seeking datasets, including NQ (Kwiatkowski et al., 2019), TriviaQA (TQA) (Joshi et al., 2017), zsRE (Levy et al., 2017), T-rex (Elsahar et al., 2018), and HotpotQA (HQA) (Yang et al., 2018), using the versions provided in KILT (Petroni et al., 2021). In all experiments, we keep the external context frozen, as the focus of this work is on evaluating the language model itself. We use either the gold contexts annotated in KILT or the top-20 passages retrieved by contriever-msmarco (Izacard et al., 2021), a strong dense retrieval model. We also include DROP (Dua et al., 2019), SQuAD (Rajpurkar et al., 2016), SWDE (Lockard et al., 2019), and FDA (Arora et al., 2023), for which we use the version curated by Based (Arora et al., 2024). Additionally, we evaluate on two benchmarks specifically designed to highlight grounding failures in language models: NQ Conflict (NQ-C) (Zhou et al., 2023) and the dataset from CORG (Lee et al., 2025)⁷. For both training and evaluation dataset, we used English dataset.

For the dataset from CORG, we report the D-F1 metric introduced in their work, which measures whether the generated response contains a disambiguated correct answer. For the other datasets, we

⁷The datasets used in our experiments are released under the following licenses: Natural Questions (NQ), SWDE, and NQ Conflict (NQ-C) under the Creative Commons Attribution 4.0 (CC BY 4.0) license; TriviaQA (TQA), T-REx, HotpotQA (HQA), DROP, and SQuAD under the Creative Commons Attribution-ShareAlike 4.0 (CC BY-SA 4.0) license; zsRE and CORG under the MIT License; and FDA under the Apache License 2.0.

evaluate using answer accuracy, which measures whether the correct answer appears in the generated response.

We further evaluate with 7 downstream tasks (PIQA (Bisk et al., 2020), Social IQa (Sap et al., 2019), Winogrande (Sakaguchi et al., 2019), HellaSwag (Zellers et al., 2019), LAMBADA-OpenAI (Paperno et al., 2016), ARC-Challenge, and ARC-Easy (Clark et al., 2018)) to evaluate overall language model performance through lm-evaluation-harness (Gao et al., 2024). We report normalized answer scores for all tasks except ARC-Easy, for which we use answer accuracy.

A.3 Visual Language Domain

Baseline We compare vision-language models that use a language model trained on context-free data (NOCTX-LLM) versus context-augmented data (CTX-LLM) as the LLM backbone for vision language alignment, resulting in NOCTX-VLM and CTX-VLM, respectively. The backbone LLMs are the same as those evaluated in Section A.2. For all models, we adopt a widely used two-stage vision-language training pipeline (Chen et al., 2024; Liu et al., 2023b): (1) a pretraining stage for feature alignment, and (2) an end-to-end fine-tuning stage. Note that the vision-language alignment procedure is identical across all experiments, and only the backbone language model varies.

Datasets & Evaluation Metrics When training the vision-language alignment, we use the training dataset from LLaVA (Liu et al., 2023b) for both the pretraining and finetuning stages; filtered CC-595K subset for pretraining and LLaVA-Instruct-158K for finetuning. To evaluate how well each model grounds its responses to the provided image, we mainly conduct experiments on four benchmarks commonly used to measure hallucination in vision-language models: AMBER (Wang et al., 2023), POPE (Li et al., 2023b), ImageInWords (Garg et al., 2024), and LLaVA-Wild (Liu et al., 2023b). For POPE, we report the average F1 score across all splits (popular, adversarial, and random). For AMBER, we evaluate performance on both generative and discriminative tasks. For ImageInWords, we adopt the evaluation metric from CapMAS (Lee et al., 2024b), which uses a GPT-based method to assess factuality in fine-grained manner. For LLaVA-Wild, we use the general GPT4-Eval metric. We adjust the evaluation rubric to explicitly pe-

nalize hallucinations, focusing the evaluation more precisely on hallucination detection. To assess overall model capabilities beyond hallucination, we additionally evaluate on four widely used downstream benchmarks: MMBench (Liu et al., 2023d), ScienceQA (Lu et al., 2022), MME (Sun et al., 2023a), and GQA (Hudson and Manning, 2019).

Training Details All models are trained using the same vision-language alignment procedure, with the only difference being the choice of language model backbone. We follow the training setup introduced in LLaVA (Liu et al., 2023b), and conduct all experiments on 8 NVIDIA A100 GPUs. During the pretraining stage, the language model is kept frozen and only the projection layer, which maps image features to language model’s word embedding space, is trained. The pretraining stage is run for one epoch with a learning rate of $2e-3$ and a batch size of 128. In the subsequent fine-tuning stage, both the projection layer and the language model are updated, while the vision encoder remains frozen throughout. Fine-tuning is performed for three epochs using a learning rate of $2e-5$ and a batch size of 32. For the vision encoder, we use the pre-trained CLIP visual encoder ViT-L/14 (Radford et al., 2021) following previous works (Liu et al., 2023b; Sun et al., 2023b).

B RQ1: How does training an LLM on instruction tuning instruction tuning with instances containing context differ from tuning without context?

B.1 Effect of number of contexts on performance

Figure 9 shows the average performance across five datasets (NQ, TriviaQA, zsRE, T-REx, and HotpotQA) (y-axis) as the number of retrieved context increases (x-axis) for CTX-LLM and NOCTX-LLM, using LLaMA 3.1 8B as the base model. We observe that CTX-LLM consistently outperforms NOCTX-LLM, but the performance gap narrows as more contexts are added. We hypothesize that this is because CTX-LLM encourages strong grounding to the provided context; therefore, when many potentially distracting contexts are present, the model may become susceptible to being misled by irrelevant information.

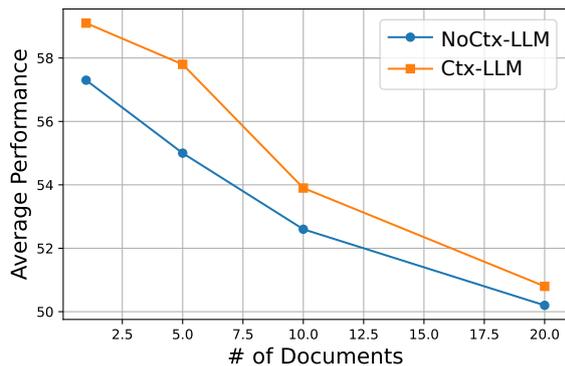


Figure 9: Avg. accuracy across nine datasets (y-axis), NQ, TriviaQA, zsRE, T-REx, and HotpotQA, as a function of the number of contexts for each instance (x-axis), for models trained with context (CTX-LLM) and without context (NOCTX-LLM).

B.2 Attention Patterns Differ Between NOCTX-LLM and CTX-LLM

Figure 10 presents attention maps during response generation for NOCTX-LLM (top) and CTX-LLM (bottom). CTX-LLM display stronger attention to the input context, whereas NOCTX-LLM attend more heavily to previously generated tokens (highlighted in the red box). This suggests that training with context-augmented data (CTX-LLM) encourages models to remain more grounded in the input, rather than relying on self-generated content.

B.3 Counterfactual Dataset Construction

In this section, we describe our procedure for constructing the counterfactual dataset, which are datasets that contains knowledge that counterfactuals with model’s prior knowledge.

Dataset Construction & Validation We categorize the dataset into two groups based on the format of the original answers: True/False (T/F) and Free-Form. For T/F examples, where the answer is either true or false, we generate counterfactuals by simply inverting the original boolean value. For Free-Form examples, which comprise the remainder of the dataset, we prompt the model to generate a counterfactual response using the template shown in Figure 11. We discard the samples for which GPT-4o⁸ refuses generation or fails to match the required format.

Using the generated counterfactual answer, we then ask GPT-4o (in the same session) to fabricate

⁸<https://openai.com/index/hello-gpt-4o/>

a supporting “background reference” using the template in Figure 12.

To ensure that each fake pair (generated counterfactual answer and external-knowledge) remains coherent, we re-initialize a fresh GPT-4o chat and validate with the template in Figure 13. Only samples for which GPT-4o selects Answer: B are retained.

Evaluation Dataset Construction We randomly sample 1k examples from the dataset and use GPT-5 to generate cloze-style questions, which are then used to evaluate whether the model knows the corresponding knowledge. Template we used to construct the evaluation dataset is in Figure 14.

Dataset Statistics and Cost The overall validation pass rates are: **T/F**: 95.5%; **Free-Form**: 97.9%. In total, we obtain 29k validated counterfactual examples (Free-Form: 24k; T/F: 5k), at a generation cost of USD \$77.83.

Results As shown in Figure 4, NoCtx(CF) shows a large drop when tested on original knowledge (Ori-Ctx) but a strong improvement when tested on counterfactual knowledge (CF-Ctx), indicating greater forgetting of prior knowledge and strong memorization of the updated (counterfactual) training signal. In contrast, models trained with context retain more of their original knowledge but memorizes counterfactual knowledge less strongly: Ctx(CF) outperforms NoCtx(CF) on Ori-Ctx, but scoring lower on CF-Ctx. Ctx models also consistently achieve higher performance than NoCtx models whenever context is available (*+Ctx), regardless of whether the context supports original or counterfactual answers. These findings suggest that context-augmented training shifts the model’s reliance from parametric memory toward external context.

C RQ2: How does using CTX-LLM or NOCTX-LLM as the backbone for vision-language adaptation influence performance on vision-language tasks?

C.1 CTX-VLM show robust performance across varying response lengths

Figure 16 presents a qualitative comparison of fine-grained captions generated by NOCTX-VLM and CTX-VLM for an instance from the ImageInWords dataset. The NOCTX-VLM tends to hallucinate, incorrectly stating that “there are two people in

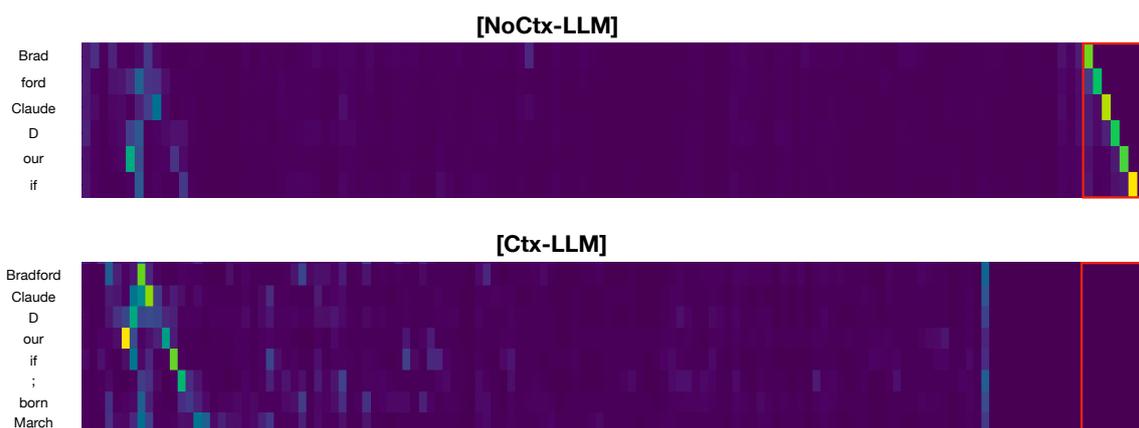


Figure 10: Attention maps during response generation for models trained with NOCTX-LLM (top) and with CTX-LLM (bottom)

Fake Answer Generation Prompt Template

You are tasked to create a binary-choice question by creating an alternative wrong answer to the provided question.

Query: {query}

Ground Truth Answer: {answer}

Create a plausible wrong answer for the provided question. Your response should be in the format of the following:

Wrong Answer: <Plausible Wrong Answer>

Figure 11: Template for Generating Fake Answers.

the scene” (red text), likely due to internal priors about humans typically playing arcade games. In contrast, CTX-VLM avoids this hallucination and instead provides a more grounded and descriptive observation (blue text), accurately noting the presence of specific buttons on the arcade machine. We observe similar trends across various other examples.

Effect of Rubric Modification on LLaVA-Wild Performance To better assess whether the model hallucinates, we modify the evaluation rubric to impose an additional penalty for hallucinated content: “Responses must remain grounded in the input image. Any hallucinated details should be heavily penalized.” With this revised rubric, we observe that the performance gap between the NOCTX-VLM and CTX-VLM increases significantly: from an average difference of 8.8% under the original rubric

to 18.1% with the modified one. This suggests that a substantial portion of the performance improvement with CTX-VLM compared to NOCTX-VLM comes from reducing hallucinations; CTX-VLM generates more factual, image-grounded responses.

Effect of context configuration in LLM instruction tuning on vision-language hallucination

We evaluate the effect of context configuration during LLM instruction tuning on vision-language adaptation. Building on the models trained on various context configurations in Section A.1, we use these LLMs as backbones for the vision-language alignment and assess their performance on vision-language hallucination benchmarks.

As shown in Table 6, VLM using LLM trained without computing loss on the context (Ctx Presence = T, Ctx Loss = F) as backbone consistently outperforms both VLM using LLMs trained with

Fake External-Knowledge Generation Prompt Template

Now, create a background reference from Wikipedia that supports your generated wrong answer. Keep the length of the reference around 100 words. Remember, your generated fictional reference should be convincing as possible so that people will be tempted to choose your generated wrong answer, ****instead of the original ground truth answer!**** The generated reference passage should seem like an excerpt from Wikipedia. This means that the reference passage should NOT start with 'According to ...'. You must NOT mention the original answer in your new reference passage. Answer in the format of the following:

Reference Passage (Around 100 Words): <Fictional Passage>

Figure 12: Template for Generating Fake External-Knowledge.

Validation Prompt Template

For the following query, pick the answer which is directly supported by the provided passage from Wikipedia. The correctness of the answer does not matter. Focus only on which answer the reference passage directly supports.

Query: {query}

Reference Passage: {fake_reference}

Choice of Answers:
 A: {original_answer}
 B: {fake_answer}

Provide your answer in the following format:

Answer: <A or B>

Figure 13: Template for Validation.

Table 6: Performance on VLM hallucination benchmarks using LLMs trained under different context configurations during instruction tuning.

Ctx Configuration		Pope	Amber						Llava-W	ImageInWords
Ctx Presence	Ctx Loss	F1	CHAIR (↓)	Cover (↑)	Hal (↑)	Cog (↓)	F1 (↑)	AMBER (↑)	GPT4-Eval	F1
F	-	87.3	9.1	53.8	39.8	5.9	66.5	78.7	55.7	54.5
T	T	87.3	8.6	53.4	37.0	4.8	70.6	81.0	62.7	55.1
T	F	87.7	8.6	54.6	47.2	4.9	71.7	81.6	74.2	56.8

loss on the context (Ctx Presence = T, Ctx Loss = T) and those trained without any added context (Ctx Presence = F) as backbone. This pattern mirrors our findings in the text domain (Section A.1) and provides evidence that the context configuration used during instruction tuning influences how models utilize and source knowledge, ultimately affecting downstream performance.

Cloze-Style Question Generation

You are given a passage and a target answer. Your task is to create a cloze-style questions where the blank should be filled by the given answer and placed at the end of the sentence. Avoid simply copying the passage verbatim; rewrite or paraphrase so it feels like a natural question.

Context: {context}

Ground Truth Answer: {answer}

Figure 14: Template for Generating Cloze-style Questions.

Table 7: Performance difference between using the original Llama-Wild rubric (Original) and a modified version that imposes stronger penalties for hallucinations (Changed).

	Llama2 (7B)		Llama3.1 (8B)		Qwen2.5 (7B)	
Original	66.3	53.4	67.3	55.7	69.6	60.1
Changed	72.1	70.9	77.1	74.2	80.1	78.5

D RQ3: How can these insights guide when to use each model and combine them effectively for downstream applications?

D.1 Using CTX-LLM over NOCTX-LLM is complementary with inference-time grounding techniques

To examine the practical application and benefits of our analysis, we evaluate whether CTX-LLM further improves grounding performance over NOCTX-LLM when used as the LLM for inference-time grounding approaches. We experiment over two inference-time grounding approaches: AdaCAD (Wang et al., 2024a), a decoding-based approach that improves grounding by adjusting the output distribution through logit weighting between parametric and contextual knowledge; and CORG (Lee et al., 2025), a pipeline-based framework designed for contexts involving complex, interrelated facts—settings where language models often struggle.

Table 9 presents the performance of NOCTX-LLM and CTX-LLM when combined with inference-time grounding techniques. We find that using CTX-LLM as the LLM for such methods consistently improves performance across all settings compared to NOCTX-LLM. For example,

using CTX-LLM with AdaCAD yields an average absolute improvement of 5.3 points and pairing it with CORG gives a 6.5 point gain compared to using NOCTX-LLM as the LLM.

D.2 Routing Input Based on Context Availability using LoRA

LoRA-based training shows trends consistent with full-parameter training Table 8 shows the performance of models trained with LoRA (Hu et al., 2022) using either context-free instruction tuning data (NOCTX-LLM) and context-augmented data (CTX-LLM), with Llama3.1 8B as the base model. For the experiment, we trained only the LoRA parameters, using a rank of 16, an alpha of 32, and a dropout rate of 0.05. The trend tends to be similar with when training the full parameter; CTX-LLM consistently outperforms NOCTX-LLM, with a larger performance gain on NQ-C and CORG (+8.9%) compared to the others (+2.0%).

Routing Inputs Based on Context Availability

Figure 15 shows a trend similar to that observed with full parameter training. *Routing* provides a strong balance, achieving robust performance in both settings, an average of 54.5, compared to 50.2 for NOCTX-LLM and 52.9 for CTX-LLM. Combining the two datasets (*Combine*) also yields a good balance, with an average of 53.0. These results suggest that when full-parameter training is computationally expensive, LoRA-based routing is an efficient alternative, especially for scaling to larger models.

Table 8: Performance across 11 information-seeking datasets using models trained with CTX-LLM and NoCTX-LLM, applied via LoRA.

Model	Training	NQ	TQA	zsRE	T-rex	HotpotQA	NQ-C	Corg	Drop	Squad	SWDE	FDA
LLaMA 3.1 8B	NoCTX-LLM	41.9	63.1	53.0	58.1	39.8	49.9	10.2	35.1	57.3	89.1	70.3
	CTX-LLM	44.2	66.8	57.6	57.9	43.1	64.1	13.7	40.6	55.9	92.7	73.2
Qwen 2.5 14B	NoCTX-LLM	49.8	76.4	74.1	75.4	39.7	60.5	20.1	52.0	58.2	81.7	82.4
	CTX-LLM	58.4	86.3	75.4	79.8	45.7	79.5	21.5	53.2	56.5	81.9	87.5
Qwen 2.5 32B	NoCTX-LLM	52.7	80.5	77.4	81.8	43.8	65.2	26.8	59.2	67.3	88.5	86.7
	CTX-LLM	61.1	88.1	78.0	84.7	48.1	83.1	30.1	61.8	69.1	91.2	92.8

Table 9: Performance over 11 information-seeking datasets using inference-based methods with NoCTX-LLM and CTX-LLM trained with Llama 2 7B or Llama 3.1 8B as base models. TQA and HQA refer to TriviaQA and HotpotQA, respectively.

Method	NQ	TQA	zsRE	T-rex	HQA	NQ-C	Corg	Drop	Squad	SWDE	FDA
Llama2 7B											
AdaCAD + NoCTX-LLM	44.9	71.4	60.2	58.0	48.8	35.9	13.8	34.4	54.5	81.3	77.0
AdaCAD + CTX-LLM	54.2	75.0	67.7	62.4	51.5	73.3	15.4	39.8	55.6	91.7	70.7
CORG + NoCTX-LLM	42.3	69.0	51.2	69.8	45.5	54.1	22.0	33.8	42.9	82.3	73.4
CORG + CTX-LLM	55.8	72.1	65.1	60.2	49.0	75.1	24.1	38.7	58.3	81.0	76.3
Llama3.1 8B											
AdaCAD + NoCTX-LLM	49.7	67.5	63.6	62.1	49.7	70.7	14.2	39.0	56.8	94.3	79.9
AdaCAD + CTX-LLM	51.7	70.1	65.8	63.9	54.3	84.7	15.9	44.3	56.9	93.0	82.9
CORG + NoCTX-LLM	48.0	70.2	57.5	62.7	48.0	60.7	21.4	39.0	64.9	92.0	74.0
CORG + CTX-LLM	46.2	72.6	62.6	63.4	50.6	72.3	28.4	44.0	69.5	95.0	80.9

D.3 Effect of varying the ratio of context-augmented and context-free training examples

Figure 7 shows the performance of models trained with the same total number of training examples but with different proportions of context-augmented data. Performance when inference is done with relevant context added (*with context*) increases as the proportion of context-augmented training examples grows. 50% mix provides the best overall balance, maintaining strong performance both with and without context.

E Checklist

E.1 Potential Risk

A model with strong grounding ability may also reliably ground on incorrect or harmful context, potentially amplifying misinformation if the provided evidence is flawed. However, we expect this risk can be mitigated by applying robust filtering and validation of external context before it is supplied to the model.

E.2 LLM Usage

We used the free version of ChatGPT-4o to assist with grammar checking during the writing of this paper.

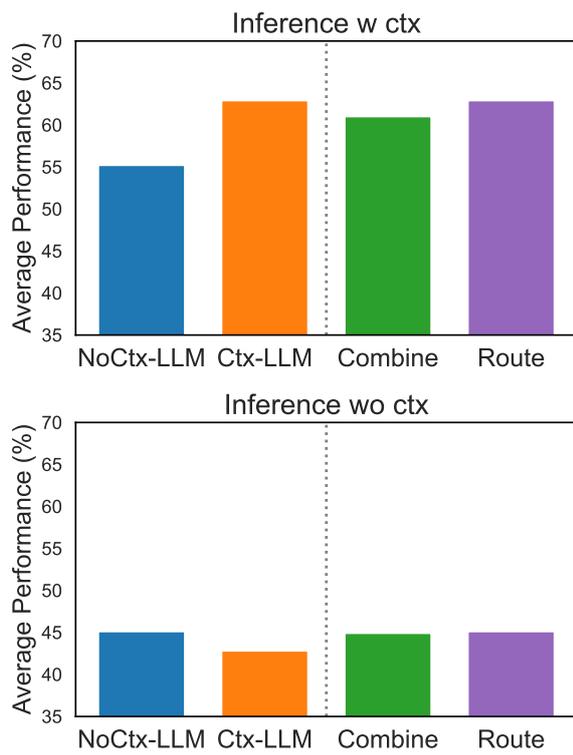


Figure 15: Avg. performance across nine datasets (NQ, TQA, zsRE, T-rer, HQA, Drop, Squad, SWDE, and FDA). Inference *w ctx* (top figure) shows performance with relevant context provided at inference time; the Inference *wo ctx* (bottom figure) shows performance without relevant context.

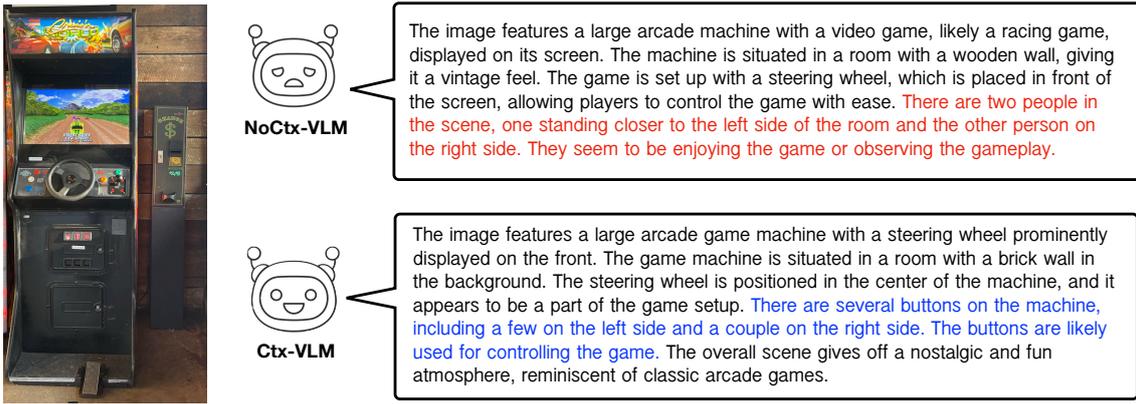


Figure 16: Example from ImageInWords. Fine-grained captions generated for the figure on the left by NOCTX-VLM and CTX-VLM.

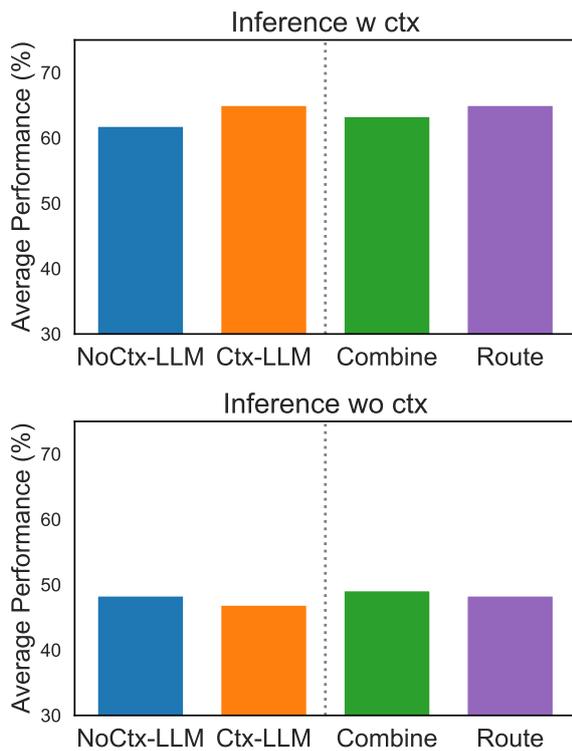


Figure 17: Avg. performance across nine datasets (NQ, TQA, zsRE, T-rer, HQA, Drop, Squad, SWDE, and FDA). Inference *w ctx* (top figure) shows performance with relevant context provided at inference time; the Inference *wo ctx* (bottom figure) shows performance without relevant context.