UNVEILING AND MITIGATING SHORT-CUT LEARNING IN MULTIMODAL IN-CONTEXT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

The performance of Large Vision-Language Models (LVLMs) in In-Context Learning (ICL) is heavily influenced by short-cut learning, particularly in tasks requiring cross-modal reasoning and open-ended generation. To address this challenge, We introduce task mapping as a novel perspective to analyze short-cut learning, revealing how existing ICD selection methods disrupt reasoning coherence. Based on this theoretical framework, we propose Task-aware model for ICL (Ta-ICL), which optimizes task mapping cohesion through task-aware attention and autoregressive retrieval. Experiments on multiple Vision-Language tasks demonstrate that Ta-ICL significantly reduces short-cut learning, enhances reasoning consistency, and improves LVLM adaptability. Our results highlight the potential of task mapping to be widely applied in enhancing multimodal reasoning, paving the way for robust and generalizable multimodal ICL frameworks.

1 INTRODUCTION

As Large Language Models (LLMs) scale up, they have demonstrated the ability to adapt to novel tasks through In-Context Learning (ICL), which uses only a few-shot forward-pass with input examples without any parameter updates (Brown et al., 2020; Lester et al., 2021; Liu et al., 2021). This efficient and low-cost learning paradigm has achieved remarkable success in LLMs (Olsson et al., 2022; Garg et al., 2023) and has since been extended to the multimodal domain. To enable multimodal ICL, some Large Vision-Language Models (LVLMs), such as Flamingo (Alayrac et al., 2022), have been designed with tailored training methods. Meanwhile, general-purpose LVLMs like InternVL2 (Chen et al., 2024) and Qwen2VL (Wang et al., 2024) have evolved to support multi-image input and reasoning, marking multimodal ICL as an essential capability for modern LVLMs.

However, with the growing application of ICL in Vision-Language (VL) tasks, certain challenges have become increasingly evident (Li et al., 2024). A major issue is short-cut learning, where models rely on spurious correlations in examples rather than genuinely understanding task mappings (Yuan et al., 2024). This challenge is closely related to the sensitivity of ICL to the selection, ordering and format of In-Context Demonstrations (ICDs) (Gao et al., 2021; Lu et al., 2022). Multimodal ICDs amplify this problem by introducing modality misalignment and task-irrelevant biases, placing higher demands on configuring ICD sequences effectively. In this work, we address two key questions to develop an ICD sequence configuration method that effectively mitigates short-cut learning:

How can we analyze the reasoning mechanism of LVLMs to uncover the root causes of short-cut learning? (§2) To better understand this issue, we introduce task mapping, which formalizes how ICDs establish a relationship between input (image, query) pairs and their expected responses. Ideally, ICDs should contribute to a cohesive task mapping, where local mappings within ICDs collectively align with the model's reasoning for the query task. However, we find that many LVLMs struggle with task mapping cohesion, leading to fragmented or misaligned reasoning. Our quantitative analysis reveals that existing ICD selection methods may disrupt task mapping cohesion, reinforcing short-cut behaviors instead of meaningful multimodal reasoning.

How can we design an ICD sequence configuration method that effectively leverages task mapping? (§3) To address these challenges, we propose Task-aware model for ICL (Ta-ICL), a novel ICD sequence configuration method that explicitly optimizes task mapping cohesion. Ta-ICL employs an autoregressive retrieval strategy to construct ICD sequences that enhance LVLM reasoning by maintaining coherent task mappings. Unlike conventional similarity-based methods, Ta-ICL inte-

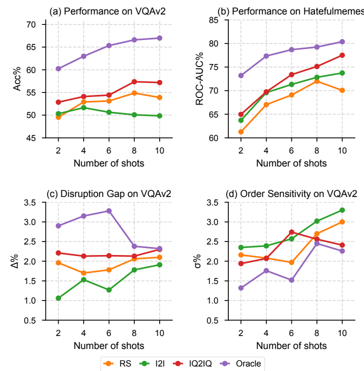

Figure 1: (a-b) Results of different ICD sequence configuration methods on VQAv2 and Hatefulmemes. (c-d) Task mapping cohesion analysis of different ICD sequence configuration methods on VQAv2.

grates task-aware attention to prioritize ICDs that contribute to a structured and contextually relevant reasoning process. Experiments across multiple VL tasks demonstrate that Ta-ICL significantly reduces short-cut learning, improving both accuracy and robustness in multimodal ICL.

1.1 Why short-cut learning happens in multimodal ICL?

1.1.1 Multimodal ICL creates task mappings.

In this work, we focus mainly on ICL for image-to-text tasks, where ICD sequences are organized in an interleaved image-text format. Toward a unified template for various tasks, we reformat ICDs as triplets (I, Q, R), where I is an image, Q is a task-specific text query and R is the ground-truth result. The query sample is denoted as $(\hat{I}, \hat{Q})$. Formally, ICL can be represented as:

$$\hat{R} \leftarrow \mathcal{M}(S^n) = \mathcal{M}(Inst; \underbrace{(I_1, Q_1, R_1), ..., (I_n, Q_n, R_n)}_{n \times ICDs}; (\hat{I}, \hat{Q})), \tag{1}$$

where $\mathcal{M}$ is a pretrained LVLM, S^n is an ICD sequence consists of an instruction $Inst$, n-shot ICDs and a query sample.

We formalize task mapping as follows: each ICD (I_i, Q_i, R_i) defines a local mapping:

$$f_i : (I_i, Q_i) \rightarrow R_i, i = 1, 2, ..., n, \tag{2}$$

and the model tries to synthesize these mappings to establish a global mapping for the query sample:

$$\hat{f} : (\hat{I}, \hat{Q}) \rightarrow \hat{R}. \tag{3}$$

We categorize multimodal ICL tasks into two types according to the heterogeneity of local task mappings: specific-mapping tasks and generalized-mapping tasks. In the former, all ICD mappings f_i converge on a focused mapping f, which also aligns with $\hat{f}$. This often applies to tasks that are novel to the LVLM or require more complex reasoning steps. In this work, we focus on the later, where ICDs' f_i exhibit fine-grained or more general differences, so it is difficult to directly unify them into $\hat{f}$. This type of task is more closely aligns with real-world scenarios, and empirical findings indicate that short-cut learning is most prevalent in it. We turn to sequence-level study and demonstrate with an open-ended VQA dataset VQAv2 (Goyal et al., 2017).

1.1.2 Multimodal ICL needs task mapping cohesion.

Three configuration methods are evaluated: Random Sampling (**RS**), similarity-based retrieval, and **Oracle**. Similarity-based retrieval selects top-n ICDs using CLIP-based cosine similarity, either via **I2I** (image-only alignment) or **IQ2IQ** (joint image-query alignment). The idealized **Oracle**

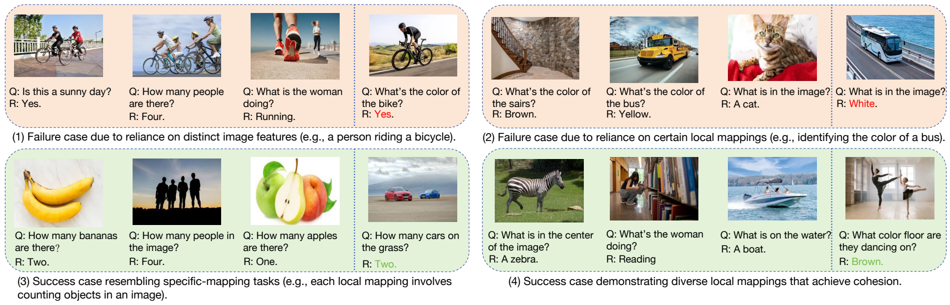

Figure 2: Four types of ICD sequences in generalized-mapping task. The first two types exhibit clear signs of short-cut learning.

method iteratively selects the next ICD by maximizing the log-likelihood of generating the ground-truth $\hat{R}$ while accounting for the cohesive influence of preceding ICDs (computational details in Appendix A.1). This greedy method goes beyond feature matching, though its reliance on $\hat{R}$ makes it impractical for real-world use.

Figure 4(a-b) shows that multimodal alignment (IQ2IQ) consistently outperforms unimodal (I2I) and random (RS) methods across tasks, with **Oracle** achieving peak performance. A key anomaly is that I2I underperforms RS in VQAv2 but not in HatefulMemes. We attribute this divergence to **task mapping cohesion**—generalized-mapping tasks (e.g., VQAv2) demand ICD sequences that collectively resolve interdependent multimodal logic. Static methods like I2I, focused on isolated feature matching, disrupt cohesion and result in short-cut learning.

To validate this hypothesis, we evaluate task mapping cohesion using two metrics: Disruption Gap (Δ) and Order Sensitivity (σ) (details in Appendix A.2). These metrics reflect the impact of cohesive task mapping on multimodal ICL, with higher Δ and lower σ indicating stronger reliance on cohesive task mapping. Figure 4(c-d) shows that **Oracle** achieves the highest Δ and lowest σ across all shots, proving its ability to construct cohesive sequences through holistic consideration of preceding ICDs. However, as shots increase to 8 and 10, **Oracle**'s Δ surges while σ plunges, revealing potential local optimization issues and accumulated bias in longer sequences. Meanwhile, I2I consistently underperforms RS on both metrics, while IQ2IQ surpasses RS but remains unstable, aligning with accuracy trends in generalized-mapping tasks and supporting our hypothesis.

Finally, based on performance, Δ and σ, we identify four types of sequence, cases provided in Figure 2 (1)-(2) sequences impaired by isolated dependencies and thus showing short-cut learning (e.g., similar image features and local task mapping bias), (3) sequences resembling specific-mapping tasks, and (4) the most common type, featuring diverse local mappings that collectively enhance cohesive task mapping. This diversity enables LVLMs to overcome short-cut learning and achieve superior multimodal ICL performance.

2 THE PROPOSED METHOD.

From Section 1.1, we conclude that mitigating short-cut learning in multimodal ICL requires ensuring that ICD sequences exhibit reasonable and effective diversity during configuration. This encourages LVLMs to leverage cohesive task mapping for deeper reasoning. Since static methods struggle to integrate task mapping into the configuration process, we explore the use of a decoder-only model. Figure 3 illustrates the pipeline of Ta-ICL, which is specifically designed to select ICDs from a demonstration library DL and organize them into sequences in an autoregressive way. Ta-ICL is centered around four Transformer decoder blocks. Its vocabulary is entirely composed of samples rather than single words. All tokens correspond one-to-one with each complete sample in DL. Consequently, given a query sample as input, Ta-ICL can progressively retrieve n samples from DL based on the generated token distribution to form the optimal n-shot ICD sequence S^n.

Input Embedding. To align with the autoregressive generation process, we use two special tokens, $[BOS]$ and $[EOS]$, to mark the beginning and end of the input sequence during training. These tokens are added to Ta-ICL's vocabulary. We also introduce a $[TASK]$ token into the vocabulary

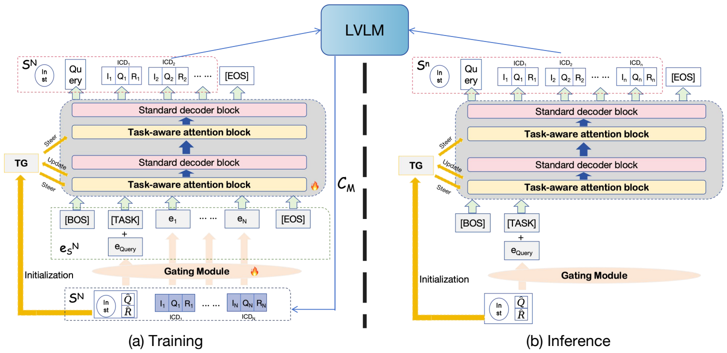

Figure 3: Overview pipeline of $SabER$.

and concatenate it with the query sample in the input sequence. It acts as a semantic anchor for task mapping, explicitly injecting task intent. In each input sequence, the query sample is placed ahead of all ICDs. Therefore, for a given sequence S^N, we reconstruct it as $\{[BOS], [TASK] + \hat{x}, x_1, ..., x_N, [EOS]\}$. To filter and balance multimodal features for deeper mapping, we employ a binary gating module to generate the embedding e_i for the i-th ICD token $x_i = (I_i, Q_i, R_i)$:

$$g_i = \sigma(W_g \cdot [E_I(I_i) \oplus E_T(Q_i \oplus R_i)] + b_g), \tag{4}$$

$$e_i = g_i \cdot E_I(I_i) + (1 - g_i) \cdot E_T(Q_i \oplus R_i), \tag{5}$$

where $E_I(\cdot)$ and $E_T(\cdot)$ denote image encoder and text encoder of CLIP. Finally, the input embedding sequence of Ta-ICL is presented as follows:

$$e_{S^N} = [e_{\text{BOS}}, \hat{e}, e_1, \ldots, e_N, e_{\text{EOS}}], \tag{6}$$

where e_{BOS} and e_{EOS} are learnable embeddings of $[BOS]$ and $[EOS]$. $\hat{e}$ is a joint representation formed by concatenating the learnable embedding of the $[TASK]$ token with the embedding of the query sample $\hat{x}$ generated using the same gating module. The index of $\hat{e}$ is always 1 and I_{idx} denotes the index set of ICD embeddings. **Task-aware Attention.** The task-aware attention in Ta-ICL enables dynamic ICD sequence configuration by integrating task mappings into attention computation. Its core is the task guider (TG), an embedding independent of the input sequence, designed to capture fine-grained global task mapping within ICD sequences. TG encodes task intent through initialization by the multimodal fusion of the query sample and instruction:

$$e_{TG}^{(0)} = W_{TG} \cdot (E_I(\hat{I}) \oplus E_T(\hat{Q}) \oplus E_T(Inst')), \tag{7}$$

where $W_{TG} \in \mathbb{R}^{d \times 3d}$ is a learnable weight matrix used to regulate the entire task guider. $Inst'$ is a simplified instruction generated by GPT-4o (Appendix B.5).

In predefined layers of task-aware attention $\mathcal{L}_T$, TG guides attention through task mapping relevance weighting. At each layer, TG interacts with token embeddings to compute relevance scores:

$$t_i^{(l)} = \sigma\Big(\text{MLP}^{(l)}(e_{TG}^{(l)} \oplus e_i)\Big), \tag{8}$$

where $\text{MLP}^{(l)}$: $\mathbb{R}^{2d} \rightarrow \mathbb{R}^d$ is a layer-specific network producing a scalar weight $g_i^l \in [0, 1]$ and σ is the sigmoid function. This weight modulates attention logits through a task-aware mask $M^{(l)}$. For intra-ICD tokens, the mask scales pairwise cosine similarities by $log(g_i^{(l)})$ to amplify task mapping cohesion. A learnable coefficient α allows the query embedding $\hat{e}$ to guide the attention throughout the sequence. Specifically, for position (i, j):

$$M_{ij}^{(l)} = \begin{cases} \frac{\text{sim}(e_i, e_j)}{\sqrt{d}} \cdot \log\big(t_i^{(l)}\big), & j \leq i \text{ and } i, j \in I_{idx}, \\ \frac{\alpha \text{sim}(\hat{e}, e_j)}{\sqrt{d}} \cdot \log\big(t_1^{(l)}\big), & i = 1 \text{ and } j \in I_{idx}, \\ -\infty, & \text{otherwise.} \end{cases} \tag{9}$$

The mask is integrated into standard attention:

$$\text{TaAttn}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}} + M^{(l)}\right) V. \tag{10}$$

TG is updated only between task-aware layers to preserve task semantic coherence, enabling hierarchical refinement from coarse task intent to fine-grained mapping. After processing layer $l \in \mathcal{L}_T$ through residual connections, TG is updated via:

$$e_{TG}^{(l')} = \mathrm{LN}\left(e_{TG}^{(l)} + \mathrm{Attention}(e_{TG}^{(l)}, H^{(l)})\right), \tag{11}$$

where l' denotes the next task-aware layer in $\mathcal{L}_T$, $H^{(l)}$ denotes the hidden states of layer l and LN denotes layer normalization. To ensure focused attention patterns, we introduce a sparsity loss that penalizes diffuse attention distributions:

$$\mathcal{L}_{\text{sparse}} = \sum_{l \in \mathcal{L}_T} \frac{1}{N} \sum_{i=1}^{N} \mathrm{KL}\left(\mathrm{softmax}(M_{i:}^{(l)}) \parallel \mathcal{U}\right), \tag{12}$$

where $\mathcal{U}$ is a uniform distribution. Minimizing this KL divergence forces the model to focus on semantically salient tokens. The total training objective combines the standard cross-entropy loss for sequence generation, sparsity regularization, and L2-norm constraint on TG to prevent overfitting:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda_1 \mathcal{L}_{\text{sparse}} + \lambda_2 \left\|W_{TG}\right\|_2^2. \tag{13}$$

Inference and Prompt Construction. After training Ta-ICL with D_S, it can autoregressively select ICDs from a library and build ICD sequences, which is detailed in Appendix B.1.

3 Experiment

3.1 Datasets and Models

We select six high-quality datasets across three key VL tasks to benchmark ICD sequences: VQAv2, VizWiz (Gurari et al., 2018), and OK-VQA (Marino et al., 2019) for open-ended VQA; Flickr30K (Young et al., 2014) and MSCOCO (Lin et al., 2014) for captioning; and HatefulMemes for classification. To further assess Ta-ICL's abilities in generalized-mapping tasks, we create a mixed-task dataset **Hybrid**, by sampling 5,000 instances from each above dataset's training set, with validation samples drawn proportionally from their validation sets. We also adopt two challenging image-to-text tasks from the latest multimodal ICL benchmark, VL-ICL (Zong et al., 2024): Fast Open-Ended MiniImageNet (**Fast**) and **CLEVR**. These tasks test whether LVLMs can capture deep task mappings from specific-mapping ICD sequences, serving as strong indicators of sequence quality. To construct the high-quality sequence dataset D_S for Ta-ICL training from the above datasets, we first reformulate them into (I, Q, R) triplets. Using clustering, we select K samples from their training sets as query samples, forming the query set $\hat{D}$. For each query sample in $\hat{D}$, N ICDs are retrieved from the remaining data using the **Oracle** method described in Section 1.1.2, creating S^N. This retrieval process is further refined through beam search to improve the quality and diversity of D_S. The implementation details are provided in Appendix B.7. All S^N begin with a CoT-style $Inst$, as detailed in *Beginning1* of Table 2.

Our experiments include four SOTA open-source LVLMs and a representative closed-source model, GPT-4V (OpenAI et al., 2024), ensuring robust evaluation. Detailed descriptions of the datasets and LVLMs are provided in Appendix C.

3.2 Baselines and Implementation Details

We adopt RS and two similarity-based retrieval methods introduced in Section 1.1.2 as baselines, as well as two additional SOTA methods.:

1. **IQPR** (Li et al., 2024): It uses RS to generate pseudo results $\hat{R}^P$, selects top-$4n$ examples based on joint similarity of I, Q, and R, and re-ranks them using Q-R similarity to obtain top-n ICDs.

2. **Lever-LM** (Yang et al., 2024): A tiny language model with four vanilla decoder layers, trained for automatic S^n configuration, serving as a key baseline.

We evaluate ICD sequences on LVLMs using validation sets of the datasets, with the training sequence shot N and the generated sequence shot n set to 4. Query set $\hat{D}$ sizes vary by dataset (Table 5). We utilize the encoders from CLIP-ViT-L/14 to generate image and text embeddings. For all

Methods	VQA			Captioning		Classification	Hybrid	Fast	CLEVR
	VQAv2 ACC.↑	VizWiz ACC.↑	OK-VQA ACC.↑	Flickr30K CIDEr↑	MSCOCO CIDEr↑	HatefulMemes ROC-AUC↑	ACC.↑	ACC.↑	ACC.↑
RS	58.79	41.94	49.89	92.02	109.26	73.00	16.85	62.66	41.51
I2I	57.21	40.58	48.57	92.94	109.65	74.02	13.00	64.49	38.63
IQ2IQ	59.88	43.81	52.13	93.00	109.75	74.37	32.40	64.47	37.37
IQPR	59.89	42.56	51.12	94.52	112.32	71.33	28.67	63.99	41.00
Lever-LM	62.31	46.83	55.10	97.48	116.90	77.94	39.29	65.02	43.66
Ours	**65.60**	**50.77**	**58.55**	**99.42**	**119.27**	**79.78**	**42.93**	**67.10**	**45.57**

Table 1: Results of different ICD sequence configuration methods across 9 datasets, with both training and generated sequences being 4-shot. Each result is the average performance across five LVLMs with the same prompt format. The highest scores are highlighted in **bold**. Underlined values indicate the results of the best baselines. Detailed results for each LVLM can be found in Figure 6.

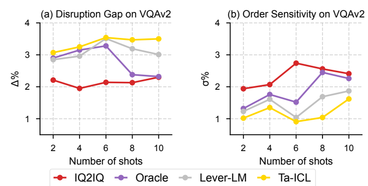

Figure 4: Analysis of task mapping cohesion in n-shot ICD sequences generated by different methods.

tasks, we employ a unified encoder training strategy: updating only the last three layers while keeping all preceding layers frozen. Ta-ICL training employs a cosine annealed warm restart learning scheduler, AdamW optimizer, 1e-4 learning rate, batch size 128, and runs for 20 epochs.

3.3 Main Results

Table 1 summarizes the average ICL performance across five LVLMs under different ICD sequence configuration methods. Ta-ICL consistently outperforms all baselines across all nine datasets, highlighting its robustness and effectiveness in fully leveraging the potential of LVLMs for diverse multimodal ICL scenarios. Notably, Ta-ICL delivers particularly strong results in generalized-mapping tasks, achieving an average improvement of 6.65% in VQA tasks, with the highest gain of 9.26% observed on **Hybrid**. These results demonstrate that strengthening task mapping enhances the autoregressive generation process of language models, equipping them with a broader understanding and enabling the creation of more precise cohesive task mappings. This results in a diverse ICD sequence, effectively mitigating the issue of short-cut learning. In Appendix C.4, we further investigate the impact of ICD sequence configuration on the LVLMs' multimodal ICL with detailed data.

3.4 Sequence-level Analyses

We again utilize the two metrics introduced in Section 1.1.2, Disruption Gap (Δ) and Order Sensitivity (σ), to evaluate task mapping cohesion in ICD sequences generated by Ta-ICL. Figure **??** shows that Ta-ICL achieves the highest Δ and lowest σ across all shots. This indicates that Ta-ICL-generated ICD sequences construct robust task mappings effectively utilized by LVLMs and mitigate short-cut learning. Notably, from the results at shots 8 and 10, we observe that although Ta-ICL's training data is constructed by **Oracle**, it overcomes the cohesion weakening caused by bias accumulation through task mapping augmentation.

4 Conclusion

This work establishes task mapping as a key concept in understanding short-cut learning in multimodal ICL. Our analysis shows that fragmented task mapping leads to unreliable reasoning, which existing ICD selection methods fail to address. To overcome this, we propose Ta-ICL, a novel approach that explicitly optimizes task mapping cohesion. Extensive experiments validate its effectiveness, showing substantial improvements in accuracy and robustness.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022. URL `https://arxiv.org/abs/2204.14198`.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL `https://arxiv.org/abs/2005.14165`.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024.

Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners, 2021. URL `https://arxiv.org/abs/2012.15723`.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes, 2023. URL `https://arxiv.org/abs/2208.01066`.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3608–3617, 2018.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning, 2021. URL `https://arxiv.org/abs/2104.08691`.

Li Li, Jiawei Peng, Huiyi Chen, Chongyang Gao, and Xu Yang. How to configure good in-context sequence for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26710–26720, 2024.

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

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, 2021. URL `https://arxiv.org/abs/2107.13586`.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity, 2022. URL `https://arxiv.org/abs/2104.08786`.

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

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads, 2022. URL https://arxiv.org/abs/2209.11895.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

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

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Yang Wu, Shilong Wang, Hao Yang, Tian Zheng, Hongbo Zhang, Yanyan Zhao, and Bing Qin. An early evaluation of gpt-4v(ision), 2023. URL https://arxiv.org/abs/2310.16534.

Xu Yang, Yingzhe Peng, Haoxuan Ma, Shuo Xu, Chi Zhang, Yucheng Han, and Hanwang Zhang. Lever lm: Configuring in-context sequence to lever large vision language models, 2024. URL https://arxiv.org/abs/2312.10104.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

Yu Yuan, Lili Zhao, Kai Zhang, Guangting Zheng, and Qi Liu. Do llms overcome shortcut learning? an evaluation of shortcut challenges in large language models, 2024. URL https://arxiv.org/abs/2410.13343.

Yongshuo Zong, Ondrej Bohdal, and Timothy Hospedales. Vl-icl bench: The devil in the details of multimodal in-context learning, 2024. URL https://arxiv.org/abs/2403.13164.

A Task Mapping

A.1 Oracle

Oracle uses the same LVLM $\mathcal{M}$ for both configuring the ICD sequences and performing ICL. This method aims to construct high-quality ICD sequences by iteratively evaluating and selecting demonstrations based on their contribution to the model's predictive performance. Given the ground-truth result $\hat{R} = (\hat{R}^{(1)}, ..., \hat{R}^{(t)})$ of the query sample, Oracle computes the log-likelihood score $\mathcal{C}_{\mathcal{M}}(S^n)$ for a sequence S^n with n ICDs, defined as:

$$\mathcal{C}_{\mathcal{M}}(S^n) = \sum_t logP_{\mathcal{M}}(\hat{R}^{(t)} \mid S^n, \hat{R}^{(1:t-1)}), \tag{14}$$

where $\mathcal{M}$ denotes the LVLM. This score measures how effectively the model predicts the ground-truth result $\hat{R}$ given the current ICD sequence S^n.

The configuration process begins with an empty sequence S^0 and iteratively selects demonstrations. At each step n, a demonstration x_n is chosen from the library D to maximize the incremental gain in the log-likelihood score:

$$x_n = \underset{x \in D}{argmax}[\mathcal{C}_{\mathcal{M}}(S^{n-1} + x) - \mathcal{C}_{\mathcal{M}}(S^{n-1})]. \tag{15}$$

This greedy optimization process ensures that each selected demonstration contributes optimally to the sequence. Unlike simple similarity-based methods, **Oracle** evaluates the overall impact of each candidate demonstration on the sequence's quality.

A.2 Task Mapping Cohesion Metrics

Δ measures performance degradation when replacing individual ICDs with another from the same sequence. σ captures performance variance under random shuffling of ICD order.

A.2.1 Disruption Gap (Δ

To measure the impact of individual ICDs on sequence-level performance and assess task mapping cohesion, we define the Disruption Gap (Δ) as the magnitude of performance change caused by replacing a single ICD in the sequence.

For each ICD $x_i = (I_i, Q_i, R_i)$ in the sequence S^n, a replacement ICD $x_j = (I_j, Q_j, R_j)$ is selected from the same dataset based on the highest joint similarity of their image and query embeddings (IQ2IQ). The modified sequence $S_{\text{replaced},i}$ is then constructed by replacing x_i with x_j.

The Disruption Gap for the i-th ICD is defined as the absolute difference in performance before and after the replacement:

$$\Delta_i = \left|\mathcal{L}(S) - \mathcal{L}(\mathcal{S}_{\text{replaced},i})\right|, \tag{16}$$

where $\mathcal{L}(\cdot)$ represents the performance metric of the sequence (e.g., accuracy).

For a sequence $\mathcal{S}$ with N ICDs, the overall Disruption Gap is computed as the average Δ_i across all N ICDs:

$$\Delta = \frac{1}{N}\sum_{i=1}^{N}\Delta_i. \tag{17}$$

To ensure the robustness of Δ and to account for potential variability in replacement effects, we conduct repeated experiments. This metric quantifies the sequence's cohesion by assessing the sensitivity of the overall performance to individual replacements. A higher Δ indicates that the sequence has stronger cohesion, as replacing an ICD results in larger performance changes.

A.2.2 Order Sensitivity (σ)

For an ICD sequence S^n, we generate K independent random permutation of it:

$$S^n_{\text{permute},1}, S^n_{\text{permute},2}, \ldots, S^n_{\text{permute},K}, \quad K = 10. \tag{18}$$

Then we compute the accuracy for each permuted sequence:

$$\text{Acc}(S^n_{\text{permute},k}) = \frac{\text{Correct Predictions}}{\text{Total Predictions}}, \quad k = 1, 2, \ldots, K. \tag{19}$$

Then calculate the mean accuracy across all permutations:

$$\mu = \frac{1}{K}\sum_{k=1}^{K}\text{Acc}(S^n_{\text{permute},k}). \tag{20}$$

Finaly, compute the standard deviation of accuracies as σ:

$$\sigma = \sqrt{\frac{1}{K}\sum_{k=1}^{K}\Big(\text{Acc}(S^n_{\text{permute},k}) - \mu\Big)^2}. \tag{21}$$

B Method

B.1 Inference and Prompt Construction

After training, Ta-ICL can autoregressively select demonstrations from a library and build ICD sequences. Given a query sample $\hat{x} = (\hat{I}, \hat{Q})$, the input sequence to Ta-ICL during inference is $\{[BOS], [TASK] + \hat{x}\}$, where $\hat{x}$ is embedded using the trained gating module. The number of ICD shots in the generated sequence, denoted as n, is a user-defined value. It may differ from the shot count N in D_S, as discussed in Section **??**. Ta-ICL then selects n ICDs using a beam search strategy with a beam size of 3, producing the optimal n-shot ICD sequence S^n. This sequence is used to construct a prompt for LVLMs, formatted as: $(Inst; ICD_1, ..., ICD_n; Query\,Sample)$, which is then used to perform multimodal ICL. Example prompts are provided in Appendix B.6.

B.2 CLIP Encoders

CLIP employs two distinct encoders: one for images and another for text. The image encoder transforms high-dimensional visual data into a compact, low-dimensional embedding space, using architectures such as a ViT. Meanwhile, the text encoder, built upon a Transformer architecture, generates rich textual representations from natural language inputs.

CLIP is trained to align the embedding spaces of images and text through a contrastive learning objective. Specifically, the model optimizes a contrastive loss that increases the cosine similarity dataset-v2.pdf

Figure 5: Illustrative examples from various vision-and-language datasets categorized by task type. Visual Question Answering (VQA) tasks are shown in red (VQAv2: train, VizWiz: laptop, OK-VQA: bus). Captioning tasks are represented in blue (Flickr30k: footbridge, MSCOCO: giraffes), while classification tasks are highlighted in green (HatefulMemes: meme identified as hateful). The bottom section demonstrates reasoning tasks with synthetic datasets: Fast Open-Ended MiniImageNet and CLEVR, focusing on conceptual understanding (e.g., assigning labels like ”Dax” or identifying object properties like color and size).

for matched image-text pairs, while reducing it for unmatched pairs within each training batch. To ensure the learning of diverse and transferable visual concepts, the CLIP team curated an extensive dataset comprising 400 million image-text pairs, allowing the model to generalize effectively across various downstream tasks.

In our experiments, we employ the same model, CLIP-ViT-L/14, using its image and text encoders to generate the image and text embeddings for each demonstration, ensuring consistency in cross-modal representations. The model employs a ViT-L/14 Transformer architecture as the image encoder and a masked self-attention Transformer as the text encoder. We experimented with several strategies for training the CLIP encoder and found that training only the last three layers of the encoder offers the best cost-effectiveness.

B.3 Demonstration Configuring Details

(a) **Open-ended VQA**: The query Q_i is the single question associated with the image I_i, while the result Ri is the answer to the question, provided as a short response. For the query sample, $\hat{Q}$ represents the question related to the image $\hat{I}$, and $\hat{R}$ is the expected output of the model.

(b) **Image Captioning**: Both Q_i and $\hat{Q}$ are set as short prompts instructing the LVLM to generate a caption for the given image, such as ”Please write a caption to describe the given image.” The result R_i corresponds to the actual caption of the image.

(c) **Image Classification**: Both Q_i and $\hat{Q}$ provide the textual information paired with the image, followed by a directive requiring the model to classify based on the provided image-text pairs. The result R_i is the predefined class label.

For all three tasks mentioned above, since the ground truth answers are not visible to the LVLM during reasoning, all $\hat{R}$ are set to blank.

B.4 Retrieving Strategies

Previous works have typically focused on calculating the similarity between either the image or parts of the textual information in the query sample and the demonstrations from the library in isolation. However, this approach can lead to insufficient use of demonstrations by the LVLM, as discussed in Section 3. To address this issue, we propose a fusion-based retrieval strategy *IQ2IQ(image-query to image-query)*, which contains two implementation methods:

(1) **Averaged Modality Similarity (AMS)** calculate the similarity between $\hat{I}$ and each I_i, and between $\hat{Q}$ and each Q_i, then take the average of these two similarities;

(2) **Joint Embedding Similarity (JES)** compute the joint image-text similarity, which concatenates the image and query embeddings to form a comprehensive vector, and use this unified representation to compute the similarity.

B.5 Instruction

The $Inst$ generated by GPT-4o in the main experiment is ”You will be provided with a series of image-text pairs as examples and a question. Your task involves two phases: first, analyze the provided image-text pairs to grasp their context and try to deeply think about what the target task is; second, use this understanding, along with a new image and your knowledge, to accurately answer the given question." This content demonstrates great orderliness and can act as a good general semantic guide for ICDs and query sample. This style is named chain-of-thought (CoT).

To incorporate the semantic information of $Inst$ and strengthen task representation during the ICD sequence configuration process, we use GPT-01 to generate simplified versions of these $Inst$ and integrate their embeddings into the task guider, which are indicated by $Inst'$. The prompt we use is as follows: *"This is an instruction to enable LVLMs to understand and perform a multimodal in-context learning task. Please simplify it by shortening the sentence while preserving its function, core meaning, and structure. The final version should be in its simplest form, where removing any word would change its core meaning"*. This simplification process allows us to investigate how the semantic information density in the instruction impacts Ta-ICL's sequence configuration ability and the performance of LVLMs in ICL. The results show that simplifying the instruction in a prompt before embedding it in the task guider significantly improves the quality of sequence generation. It also helps to avoid issues caused by too long instructions.

As shown in Table 2, we use GPT-4o to rewrite $Inst$, placing it at the middle and the end of a prompt, altering its semantic structure accordingly while keeping its CoT nature. The table also presents two other tested styles of instructions placed at the beginning of the prompt: Parallel Pattern Integration (PPI) and System-Directive (SD). PPI emphasizes simultaneous processing of pattern recognition and knowledge integration, focusing on dynamic pattern repository construction rather than sequential reasoning. SD structures input as a formal system protocol with defined parameters and execution flows, prioritizing systematic processing over step-by-step analysis. These two forms have also been proven to be effective in previous ICL work. We use them to study the robustness of Ta-ICL and various LVLMs to different instruction formats.

B.6 Prompt Details

The prompts constructed based on S^n all follow the format:

$$(Inst; ICD_1, ..., ICD_n; QuerySample).$$

Each ICD's query begins with "Question:" and its result starts with "Answer:". The query sample concludes with "Answer:", prompting the LVLM to generate a response. Depending on the input format required by different LVLMs, we may also include special tags at the beginning and end of the prompt.

Table 3 provides an overview of the prompt details used for the different models in our experiments. Each model, including OpenFlamingoV2, ICDEFICSv2, InternVL2, and Qwen2VL, employs a structured approach to engage with image-text pairs. The two-phase task requires LVLMs to first absorb information from a series of prompts before utilizing that context to answer subsequent questions related to new images. This method allows for enhanced understanding and reasoning based on prior knowledge and context, which is essential for accurate question answering in vision-and-language tasks.

B.7 Training Data Construction

Training Data Construction. (1). We apply k-means clustering based on image features to partition the dataset into k clusters. From each cluster, we select the m samples closest to the centroid, yielding a total of $K = m \times k$ samples. These form the query sample set $\hat{D}$ after removing their ground-truth results, which are stored separately in $D_{\hat{R}}$. The remaining dataset serves as the demonstration library DL. (2). For each query sample $\hat{x}_i \in \hat{D}$, we randomly sample a candidate set D_i of $64n$ demonstrations from DL. The objective is to retrieve N demonstrations from D_i that optimally configure the sequence for $\hat{x}_i = (\hat{I}_i, \hat{Q}_i)$ with its ground-truth result $\hat{R}_i = (\hat{R}_i^{(1)}, ..., \hat{R}i^{(t)})$. We use the log-likelihood score computed by the LVLM $\mathcal{M}$ as the selection criterion $\mathcal{CM}$, evaluating the model's predictive ability given a sequence with n ICDs:

$$\mathcal{C}_{\mathcal{M}}(S_i^n) = \sum_t logP_{\mathcal{M}}(\hat{R}_i^{(t)} \mid S_i^n, \hat{R}_i^{(1:t-1)}),$$

$Inst$	**Details**
Beginning1 (CoT)	You will be provided with a series of image-text pairs as examples and a question. Your task involves two phases: first, analyze the provided image-text pairs to grasp their context and try to deeply think about what the target task is; second, use this understanding, along with a new image and your knowledge, to accurately answer the given question.
Beginning2 (PPI)	Construct a dynamic pattern repository from image-text samples, then leverage this framework alongside your knowledge base for concurrent visual analysis and question resolution. The key is parallel processing - your pattern matching and knowledge integration should happen simultaneously rather than sequentially.
Beginning3 (SD)	SYSTEM DIRECTIVE Input Stream: Example Pairs → New Image + Query Process: Pattern Extract → Knowledge Merge → Visual Analysis → Response Critical: All exemplar patterns must inform final analysis Priority: Context preservation essential
Middle	Now you have seen several examples of image-text pairs. Next, you will be given a question. Your task involves two phases: first, revisit the above image-text pairs and try to deeply think about what the target task is; second, use this understanding, along with a new image and your knowledge, to accurately answer the given question.
End	Now you have seen several examples of image-text pairs and a question accompanied by a new image. Your task involves two phases: first, revisit the provided examples and try to deeply think about what the target task is; second, use this understanding, the new image and your knowledge to accurately answer the given question.
Beginning1 (Abbreviated)	Analyze the following image-text pairs, understand the task, and use this to answer the question with a new image.
Middle (Abbreviated)	After reviewing the above image-text pairs, analyze the task and use this understanding to answer the question with a new image.
End (Abbreviated)	After reviewing the above image-text pairs and a question with a new image, analyze the task and use this understanding it.

Table 2: Formats of different instruction types and their corresponding details used in the prompt structure for all VL tasks. (Abbreviated) means that the instruction is a simplified version produced by GPT-o1.

Models	**Prompt details**
OpenFlamingo-v2	Your task involves two phases: first, analyze the provided image-text pairs to grasp their context and try to deeply think about what the target task is; second, use this understanding, along with a new image and your knowledge, to accurately answer an upcoming question. < img¿<IMG_CONTEXT¿<—endofchunk—¿ Question: In what country can you see this? Answer: vietnam <img¿<IMG_CONTEXT¿<—endofchunk—¿ Question: Is this a buggy or car? Answer: buggy <img¿<IMG_CONTEXT¿<—endofchunk—¿ Question: What is this? Answer:
IDEFICSv1	”User: Your task involves two phases: first, analyze the provided image-text pairs to grasp their context and try to deeply think about what the target task is; second, use this understanding, along with a new image and your knowledge, to accurately answer an upcoming question.” ”\nUser:<—image_pad—¿ Question: In what country can you see this? <end_of_utterance¿”, ”\nAssistant: Answer: vietnam. <end_of_utterance¿”, ”\nUser: <—image_pad—¿ Question: Is this a buggy or car? <end_of_utterance¿”, ”\nAssistant: Answer: buggy. <end_of_utterance¿”, <—image_pad—¿ Question: What is this? <end_of_utterance¿”, ”\nAssistant: Answer:”
InternVL2	Your task involves two phases: first, analyze the provided image-text pairs to grasp their context; second, use this understanding, along with a new image and your knowledge, to accurately answer an upcoming question. <img¿<IMG_CONTEXT¿</img¿ Question: In what country can you see this? Answer: vietnam <img¿<IMG_CONTEXT¿</img¿ Question: Is this a buggy or car? Answer: buggy <img¿<IMG_CONTEXT¿</img¿ Question: What is this? Answer:
Qwen2VL	<—im_start—¿system You are a helpful assistant.<—im_end—¿ <—im_start—¿user Your task involves two phases: first, analyze the provided image-text pairs to grasp their context and try to deeply think about what the target task is; second, use this understanding, along with a new image and your knowledge, to accurately answer an upcoming question. <—vision_start—¿<—image_pad—¿<—vision_end—¿Question:In what country can you see this? Answer: vietnam <—vision_start—¿<—image_pad—¿<—vision_end—¿Question: Is this a buggy or car? Answer: buggy <—vision_start—¿<—image_pad—¿<—vision_end—¿Question: What is this? Answer: <—im_end—¿ <—im_start—¿assistant

Table 3: Prompt details for different models used in the experiments. The table outlines how OpenFlamingo-v2, IDEFICSv1, InternVL2, and Qwen2-VL format their image-text interactions, including examples of image-based questions and short answers. Each model follows a multi-phase task structure, where context is absorbed from previous image-text pairs to answer subsequent questions.

Datasets	VQAv2	VizWiz	OK-VQA	Flickr30k	MSCOCO	HatefulMemes	Hybrid	Fast	CLEVR
metrics	Accuracy	Accuracy	Accuracy	CIDEr	CIDEr	ROC-AUC	Accuracy	Accuracy	Accuracy

Table 4: Evaluation metrics used for each dataset. Accuracy is used for VQA datasets (VQAv2, VizWiz, OK-VQA), self-bulit **Hybrid** dataset and two VL-ICL Bench's tasks. CIDEr (Vedantam et al., 2015) is used for image captioning datasets (Flickr30k, MSCOCO). ROC-AUC is used for the HatefulMemes classification task.

Datasets	Training	Validation	Test	$\hat{D}$ Size
VQAv2	443,757	214,354	447,793	8000
VizWiz	20,523	4,319	8,000	2000
OK-VQA	9,055	5,000	/	800
Flickr30k	29,783	1,000	1,000	2500
MSCOCO	82,783	40,504	40,775	3000
HatefulMemes	8,500	500	2,000	800
Hybrid	30000	9000	/	3000
Fast	5,000	/	200	500
CLEVR	800	/	200	80

Table 5: Overview of the size distribution across the datasets used.

To determine the optimal n-th demonstration x_n for a sequence S_i^{n-1} with $n-1$ ICDs, we select the candidate that maximizes the incremental gain in $\mathcal{C}_{\mathcal{M}}$:

$$x_n = \underset{x \in D_i}{argmax}[\mathcal{C}_{\mathcal{M}}(S_i^{n-1} + x) - \mathcal{C}_{\mathcal{M}}(S_i^{n-1})].$$

(3). We employ beam search with a beam size of $2N$, ensuring that for each $\hat{x}$, the top $2N$ optimal sequences are included in D_S. As a result, the final sequence set D_S consists of $2N \times k$ N-shot sequences, providing refined training data for the model.

C Experiment

C.1 Dataset

In our study, we explore various VL tasks that use diverse datasets to evaluate model performance. As illustrated in Figure 5, we use VQA datasets such as VQAv2, VizWiz, and OK-VQA, which test the models' abilities in question-answer scenarios. Additionally, we incorporate image captioning datasets such as Flickr30k and MSCOCO to assess descriptive accuracy, along with the Hateful-Memes dataset for classification tasks focused on hate speech detection. This comprehensive approach allows us to thoroughly evaluate the models across different tasks. The size distribution of the training, validation and test sets in these VL datasets is shown in Table 5.

For the Open-ended VQA task, we utilize the following datasets: VQAv2, which contains images from the MSCOCO dataset and focuses on traditional question-answering pairs, testing the model's ability to understand both the image and the question. VizWiz presents a more challenging setting with lower-quality images and questions along with a lot of unanswerable questions, pushing models to handle uncertainty and ambiguity. OK-VQA is distinct in that it requires the model to leverage external knowledge beyond the image content itself to generate correct answers, making it a benchmark for evaluating models' capacity to integrate outside information.

For the Image Captioning task, we use the Flickr30k and MSCOCO datasets. The Flickr30k dataset consists of images depicting everyday activities, with accompanying captions that provide concise descriptions of these scenes. The MSCOCO dataset is a widely-used benchmark featuring a diverse range of images with detailed and richly descriptive captions, ideal for evaluating image captioning models.

For the Image Classification task, we use the HatefulMemes dataset, which is an innovative dataset designed to reflect real-world challenges found in internet memes. It combines both visual and textual elements, requiring the model to jointly interpret the image and the overlaid text to detect instances of hate speech.

VL-ICL Bench covers a number of tasks, which includes diverse multimodal ICL capabilities spanning concept binding, reasoning or fine-grained perception. Few-shot ICL is performed by sampling the ICDs from the training split and the query examples from the test split. We choose two image-to-text generation tasks from it, which reflects different key points of ICL. Fast Open MiniImageNet task assigns novel synthetic names (e.g., dax or perpo) to object categories, and LVLMs must learn these associations to name test images based on a few examples instead of their parametric knowledge, emphasizing the importance of rapid learning from ICDs. CLEVR Count Induction asks LVLMs to solve tasks like *"How many red objects are there in the scene?"* from examples rather than explicit prompts. The ICDs' images are accompanied by obscure queries formed as attribute-value pairs that identify a specific object type based on four attributes: size, shape, color, or material. Models must perform challenging reasoning to discern the task mapping and generate the correct count of objects that match the query attribute.

The datasets in our experiments are evaluated using task-specific metrics, as summarized in Table 4. For the VQA tasks, **Hybrid** dataset and VL-ICL bench's tasks, we use accuracy as the metric to assess the models' ability to provide correct answers:

$$Acc_{a_i} = max(1, \frac{3 \times \sum_{k \in [0,9]} match(a_i, g_k)}{10}),$$

where a_i denotes the model's generated answer, g_k denotes the k-th ground true answer. $match(\cdot, \cdot)$ decides whether two answers match, if they match, the result is 1, otherwise it is 0.

For the image captioning tasks, we use the CIDEr score, which measures the similarity between generated captions and human annotations. Finally, for the HatefulMemes classification task, we evaluate performance using the ROC-AUC metric, which reflects the model's ability to distinguish between hateful and non-hateful content.

C.2 LVLMs

In recent advances of large vision language models (LVLMs), efficient processing of multimodal inputs, especially images, has become a critical focus. Models like OpenFlamingoV2, IDEFICSv2, InternVL2, Qwen2-VL and GPT-4V implement unique strategies to manage and process visual data alongside textual input.

OpenFlamingoV2 handles visual input by dividing images into patches and encoding them with a Vision Transformer. Each image patch generates a number of visual tokens, which are then processed alongside text inputs for multimodal tasks. To manage multi-image inputs, the model inserts special tokens <image¿ and <—endofchunk—¿ at the beginning and end of the visual token sequences. For example, an image divided into 4 patches produces 4 x 256 visual tokens, with the additional special tokens marking the boundaries before the tokens are processed by the large language model.

IDEFICS2 processes visual input by applying an adaptive patch division strategy adapted to image resolution and content complexity. Depending on these factors, each image is segmented into 1 to 6 patches, striking a balance between preserving spatial information and maintaining efficiency. These patches are encoded through a Vision Transformer, followed by a spatial attention mechanism and a compact MLP, resulting in 128 visual tokens per patch. The positions of images in the input sequence are marked with <—image_pad—¿ for alignment, while <end_of_utterance¿ tokens separate query and answer components in in-context demonstrations. An image split into five patches yields 5 x 128 + 2 tokens before being integrated with the LLM.

InternVL2 also dynamically divides images into 1 to 4 patches based on their aspect ratio. A Vision Transformer then extracts visual features from each patch, followed by a pixel shuffle operation and a mlp, producing 256 visual tokens for each patch. Additionally, special tokens <img¿ and </img¿ are inserted at the beginning and end of the sequence. So, an image divided into 3 patches will produce 3 x 256 + 2 tokens before entering LLM.

Qwen2-VL reduces the number of visual tokens per image through a compression mechanism that condenses adjacent tokens. A ViT first encodes an image (e.g., with a resolution of 224 x 224 and a patch size of 14), producing a grid of tokens, which is then reduced by employing a simple MLP to compress 2 x 2 tokens into a single token. Special <lvision_start—¿ and <lvision_end—¿ tokens are inserted at the start and end of the compressed visual token sequence. For example, an image that initially generates 256 visual tokens is compressed to just 66 tokens before entering the LLM.

GPT-4V (Vision) extends GPT-4's capabilities to handle VL tasks by enabling the model to process and reason about visual input alongside text. The model can perform various tasks including image understanding, object recognition, text extraction, and visual question-answering through natural language interaction. In terms of its few-shot learning ability, GPT-4V demonstrates the capacity to adapt to new visual tasks given a small number of examples through natural language instructions, showing potential in areas such as image classification and visual reasoning, though performance may vary across different task domains and complexity levels.

C.3 Baseline

Various baseline methods are used to evaluate the model's performance, ranging from random sample to different SOTA retrieval strategies. The following is a description of the baselines used in our experiments.

1. **Random Sampling (RS)**: In this approach, a uniform distribution is followed to randomly sample n demonstrations from the library. These demonstrations are then directly inserted into the prompt to guide the model in answering the query.

2. **Image2Image (I2I)**: During the retrieval process, only the image embeddings I_i from each demonstration $(I_i, Q_i, R_i$ are used. These embeddings are compared to the query image embedding $\hat{I}$ and the retrieval is based on the similarity between the images.

3. **ImageQuery2ImageQuery (IQ2IQ)**: During the retrieval process, both the image embeddings I_i and the query embeddings Q_i of each demonstration $(I_i, Q_i, R_i$ are used. These embeddings are compared to the embedding of the concatenated query sample $(\hat{I}, \hat{Q})$ and the retrieval is based on the joint similarity between the images and the queries.

4. **ImageQuery&Pseudo Result (IQPR)**: This baseline starts by using the RS to generate a pseudo result $\hat{R}^P$ of the query sample. The pseudo result is then concatenated with $\hat{I}$ and $\hat{Q}$ to form the query sample's embedding. This retrieval method is based on the similarity of the whole triplet, using image, query and result embeddings.

5. **Lever-LM**: Lever-LM is designed to capture statistical patterns between ICDs for an effective ICD sequence configuration. Observing that configuring an ICD sequence resembles composing a sentence, Lever-LM leverages a temporal learning approach to identify these patterns. A special dataset of effective ICD sequences is constructed to train Lever-LM. Once trained, its performance is validated by comparing it with similarity-based retrieval methods, demonstrating its ability to capture inter-ICD patterns and enhance ICD sequence configuration for LVLMs.

C.4 Main Results

We can go deep into the results in Tabel 6. The findings are as follows: (1) Ta-ICL exhibits the best performance in all but three tasks across nine datasets and five LVLMs, demonstrating its great efficiency and generalization. Upon examining the outputs, we observe that GPT-4V tends to deviate from the ICD format and produce redundant information more easily than open-source LVLMs, aligning with (Wu et al., 2023). This results in the quality improvement of the ICD sequence not always translating into stable ICL performance gains for GPT-4V, which may explain why Ta-ICL did not achieve the best performance in two of its tasks. (2) For tasks like VizWiz and **Hybrid**, Ta-ICL consistently improves the quality of sequence generation in all LVLMs compared to similarity-based models, demonstrating the importance of increasing task semantics for complex task mappings. We find that the performance gains from Ta-ICL are not directly related to the model's intrinsic ability on these tasks. Unlike simpler tasks like captioning, for tasks with complex mappings, task semantics still has a significant impact, even when LVLMs exhibit strong few-shot learning abilities. This

		VQA			Captioning		Classification	Hybrid	Fast	CLEVR
		VQAv2	VizWiz	OK-VQA	Flickr30K	MSCOCO	HatefulMemes			
OpenFlamingov2	RS	49.52	27.71	37.90	76.74	92.98	70.53	13.48	57.69	21.60
	I2I	50.84	26.82	37.79	79.84	94.31	64.75	12.79	59.07	19.39
	IQ2IQ	52.29	31.78	42.93	79.91	94.40	68.72	24.93	58.96	20.03
	SQPR	53.38	30.12	41.70	80.02	96.37	69.16	28.71	57.32	21.84
	Lever-LM	55.89	33.34	43.65	83.17	98.74	72.70	32.04	59.41	22.67
	Ours	**60.12**	**39.76**	**46.28**	**84.23**	**99.10**	**75.09**	**35.17**	**62.25**	**26.80**
IDEFICS2	RS	53.77	32.92	40.01	82.43	99.61	68.81	15.65	54.72	35.14
	I2I	54.97	31.67	41.37	85.76	101.34	69.31	10.49	55.20	32.37
	IQ2IQ	55.41	34.31	43.13	85.63	101.45	70.78	30.36	55.14	32.75
	SQPR	55.32	33.74	42.76	87.65	103.57	62.18	24.03	55.18	36.29
	Lever-LM	56.78	34.10	43.27	88.01	105.62	71.33	30.14	55.83	38.97
	Ours	**58.41**	**38.32**	**47.35**	**90.41**	**107.04**	**73.68**	**33.25**	**61.21**	**40.21**
InternVL2	RS	61.83	54.70	57.13	99.05	116.37	76.84	17.74	75.87	57.03
	I2I	63.35	55.07	58.73	103.29	118.46	70.72	14.82	75.89	54.79
	IQ2IQ	64.57	56.94	**62.91**	103.41	118.53	78.20	36.46	76.03	50.07
	SQPR	63.67	56.83	60.14	105.28	121.94	77.31	34.05	76.34	56.32
	Lever-LM	65.36	57.27	61.11	104.65	126.12	79.58	43.16	78.84	57.45
	Ours	**68.42**	**61.69**	62.87	**108.26**	**128.34**	**82.97**	**45.79**	**81.76**	**59.27**
Qwen2VL	RS	63.71	48.97	55.30	100.32	121.47	80.01	20.42	66.29	48.70
	I2I	64.28	48.75	56.39	102.87	124.50	77.85	13.89	67.81	47.97
	IQ2IQ	67.26	52.20	58.49	103.04	124.63	79.78	37.83	67.76	46.63
	SQPR	67.49	49.54	59.86	105.13	127.38	76.67	27.96	67.12	49.56
	Lever-LM	68.23	54.81	61.75	105.24	127.03	81.29	45.47	70.73	50.85
	Ours	**71.57**	**57.93**	**63.97**	**106.91**	**132.14**	**83.19**	**48.95**	**75.09**	**55.98**
GPT-4V	RS	60.49	45.38	59.13	101.56	115.87	82.40	16.98	58.72	45.08
	I2I	-	-	-	-	-	-	-	-	-
	IQ2IQ	-	-	-	-	-	-	-	-	-
	SQPR	-	-	-	-	-	-	-	-	-
	Lever-LM	**65.31**	54.62	65.73	106.34	126.98	**84.81**	45.62	60.31	48.34
	Ours	65.16	**56.17**	**68.39**	**107.29**	**129.71**	83.96	**51.48**	**67.17**	**50.59**

Table 6: Detailed results of different methods across all tasks for the five LVLMs used in the evaluation, with all generated sequences being 4-shot. The highest scores are highlighted in **bold**. Our model achieves the best performance in all but three tasks, demonstrating its generalization and effectiveness.

shows that models with strong ICL capabilities on certain tasks retain, and even strengthen, their ability to leverage task semantics, underscoring the value of improving ICD sequence quality.