Multimodal In-context Learning Needs Task Mapping

Anonymous ACL submission

Abstract

The performance of Large Vision-Language Models (LVLMs) in In-Context Learning (ICL) is heavily influenced by the quality of ICL sequences, particularly in tasks requiring cross-modal reasoning and open-ended generation. To address this challenge, we innovatively interpret multimodal ICL from the perspective of task mapping. We systematically model local and global relationships within in-context demonstrations (ICDs) and demonstrate their core role and cohesion in enhancing LVLM performance. Inspired by these findings, we propose Ta-ICL, a lightweight transformer-based model equipped with task-aware attention to dynamically configure ICL sequences. By integrating task mapping into the autoregressive process, Ta-ICL achieves bidirectional enhancement between sequence configuration and task reasoning. Through extensive experiments, we demonstrate that Ta-ICL effectively improves multimodal ICL across various LVLMs and tasks. 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.

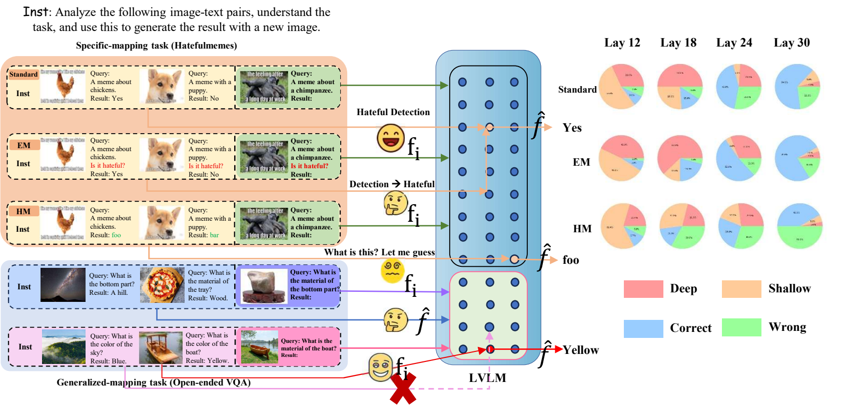
1 Introduction

As Large Language Models (LLMs) scale up, they have demonstrated remarkable adaptability to novel tasks through In-Context Learning (ICL), a paradigm that leverages a few-shot forward-pass with input examples, requiring no parameter updates (Brown et al., 2020; Lester et al., 2021; Liu et al., 2021b). This efficient and cost-effective approach has achieved notable success in LLMs (Olsson et al., 2022; Garg et al., 2023) and has since been extended to the multimodal domain. Correspondingly, Large Vision-Language Models (LVLMs) have evolved to support multi-image inputs and reasoning, establishing multimodal ICL as a crucial capability for modern LVLMs (Alayrac et al., 2022; Chen et al., 2024b).

Despite significant progress in multimodal ICL, existing studies consistently reveal that ICL performance is highly sensitive to the content and structure of the input sequence (Schwettmann et al., 2023; Zhou et al., 2024). Such sequences typically comprise an instruction, a few in-context demonstrations (ICDs), and a query sample (see Figure 1). This sensitivity highlights the critical importance of designing effective ICL sequence configuration methods. However, current approaches in vision-language (VL) tasks often struggle with complex scenarios, as they prioritize preserving data distribution rather than understanding how LVLMs internally process these sequences (Iter et al., 2023; Fan et al., 2024). This gap underscores the necessity of methods that align with the underlying mechanisms LVLMs use to synthesize information from ICDs, allowing for more effective reasoning. Furthermore, the diverse cross-modal interactions inherent in VL tasks add an additional layer of complexity, complicating efforts to develop robust sequence configuration strategies.

Towards more effective multimodal ICL, this work focuses on addressing two key questions:

How do multimodal sequences affect LVLMs' ICL performance? (§2) We introduce a novel paradigm for studying multimodal ICL: **Task Mapping**. Task mapping refers to the implicit process by which relationships between inputs and outputs in individual ICDs (local mappings) are constructed and synthesized into a cohesive framework (global mapping) to support LVLM reasoning. It captures both the latent structure within ICDs and their aggregation into a unified representation, enabling the model to navigate cross-modal interplay and deep reasoning processes. We propose a systematic framework to investigate how LVLMs utilize task mapping in ICL sequences, revealing two key findings: (1) task mapping enhances LVLM performance and plays a central role in multimodal ICL by structuring both local and global task align-


Inst: Analyze the following image-text pairs, understand the task, and use this to generate the result with a new image.
Specific-mapping task (Hatefulmemes)
Standard
Inst
Query: A meme about chickens. Result: Yes
Query: A meme with a puppy. Result: No
Query: A meme about a chimpanzee. Result:
EM
Inst
Query: A meme about chickens. Is it hateful? Result: Yes
Query: A meme with a puppy. Is it hateful? Result: No
Query: A meme about a chimpanzee. Is it hateful? Result:
HM
Inst
Query: A meme about chickens. Result: foo
Query: A meme with a puppy. Result: bar
Query: A meme about a chimpanzee. Result:
Inst
Query: What is the bottom part? Result: A hill.
Query: What is the material of the tray? Result: Wood.
Query: What is the material of the bottom part? Result:
Inst
Query: What is the color of the sky? Result: Blue.
Query: What is the color of the boat? Result: Yellow.
Query: What is the material of the boat? Result:
Generalized-mapping task (Open-ended VQA)
Hateful Detection
Detection → Hateful
What is this? Let me guess
f_i
$\hat{f}$
Yes
foo
Yellow
LVLM
Lay 12
Lay 18
Lay 24
Lay 30
Standard
EM
HM
Deep
Shallow
Correct
Wrong

(a) ICL sequences examples of specific-mapping task and generalized-mapping task.

(b) Task mappings in multimodal in-context learning.

(c) Logit lens of different settings of task mapping.

Figure 1: Examples of ICL sequences showcasing the multimodal ICL process of LVLMs based on task mapping.

ments; (2) LVLMs leverage task mapping cohesion as a critical mechanism to effectively synthesize diverse local mappings, particularly in scenarios requiring robust cross-modal reasoning. These findings highlight the importance of task mapping as a foundation for improving multimodal ICL performance and sequence configuration strategies.

How can we enhance the ICL sequence configuration for more effective task mapping? (§3) To address this, we propose Task-aware model for ICL (Ta-ICL), a lightweight model comprising a small number of Transformer decoder layers, designed to optimize ICL sequence configuration. At the core of Ta-ICL is its key innovation: a task-aware attention mechanism that dynamically incorporates task mapping into the sequence configuration process. This mechanism introduces a task guider, which encodes the latent task intent from the query and instruction, and actively steers attention computations to prioritize task-relevant features and relationships. By iteratively refining task alignment across layers, the task guider ensures a continuous flow of task information, enabling the model to effectively synthesize cohesive global mappings from diverse local mappings. This dynamic integration enhances reasoning efficiency and cohesion, allowing Ta-ICL to generate high-quality ICL sequences with strong generalization across LVLMs and VL tasks.

2 Task Mapping in Multimodal ICL

In this section, we focus on introducing task mapping. We first define task mapping (§2.1) and, through step-by-step experiments, examine its significant impact on multimodal ICL (§2.2) as well as its underlying mechanisms within LVLMs (§2.3). All experiments in this section are conducted using two LVLMs: OpenFlamingov2 (9B) (Awadalla et al., 2023) and IDEFICS2 (8B) (Laurençon et al., 2023), with the results reported as the average.

2.1 VL ICL Creates Task Mapping

In this work, we focus mainly on ICL for image-to-text tasks, where ICL 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 ICL sequence consists of an instruction $Inst$, n-shot ICDs and a query sample, as illustrated in Figure 1.

To better understand the role of ICL sequences in multimodal reasoning, we propose task mapping as a systematic framework for capturing both local and global relationships in ICL. Task mapping represents how individual input-output pairs contribute to broader reasoning and decision-making, revealing the mechanisms that LVLMs use to synthesize and align multimodal information.

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

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

and the model's generation process can be viewed as establishing a global task mapping for the query sample:

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

However, the exact role and influence of local mappings f_i in constructing the global mapping $\hat{f}$ remain unclear. To enable a systematic analysis, we first consider a uniform scenario, where all local mappings f_i converge to a focused mapping f aligned with $\hat{f}$. This commonly applies to tasks novel to the LVLM or requiring specialized reasoning. Here, I, Q, and R exhibit strong structural consistency, allowing efficient component-level analysis. We term these as **specific-mapping tasks**. Here, we demonstrate with meme review task using modified HatefulMemes (Kiela et al., 2020) (as shown in Figure 1(a)). This task requires LVLM to learn such f: detecting whether a meme image I with a short description as Q is hateful, outputting "yes" if it is; otherwise "no". We use its validation set as the query set and obtain n ICDs from its training set using Random Sampling (RS) with a normal distribution to construct sequences. We create two setups to manipulate and highlight f_i:

- *Easier Mapping (EM)*: Augment Q_i with an explicit task hint "Is it hateful?".
- *Harder Mapping (HM)*: Replace R_i (yes/no) with non-semantic words `foo/bar`.

We use logit lens (nostalgebraist, 2020) to visualize the output evolution of LVLM under three settings. We define four categories using anchor words (Appendix B.2): "Shallow" represents superficial task recognition, "Deep" indicates deeper comprehension, "Correct" corresponds to the query sample's correct answer, and "Wrong" represents the opposite. Figure 2(c) shows the probability of each layer's last token decoding to these types. The curves clearly show that *EM* greatly enhances the model's ability for deeper task recognition, while *HM* leads to a persistent lack of task awareness, causing the model to rely on guessing.

2.2 Task Mapping is Dominant

Next, we use LVLM performance to intuitively demonstrate the role of task mapping in multimodal ICL. We introduce targeted ablation settings that selectively impair label reliability and visual clarity, enabling an evaluation of whether LVLMs primarily rely on task mapping over these isolated factors. Specifically, we define:

1. **Wrong Labels (WL)**: Invert 75% R_i labels (yes↔no).
2. **Blurred Images (BI)**: Applying Gaussian blur to all I_i.

We also apply *EM* solely to $\hat{Q}$, denoted as *EM*($\hat{Q}$). *BI*($\hat{I}$) refers to applying *BI* solely to $\hat{I}$. Details are provided in Appendix B.4. ICL sequences undergo these settings and we present their performance in Figure 2. The findings are as follows.

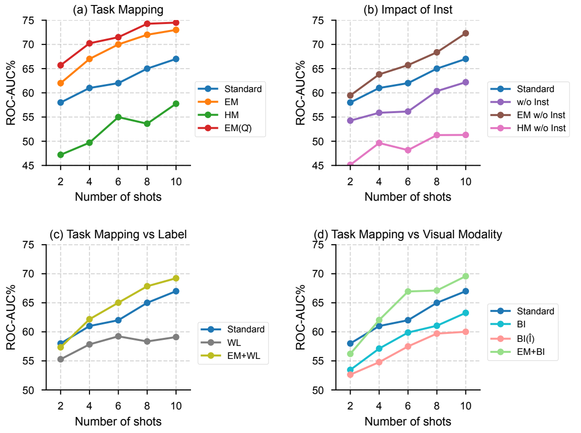


Figure 2: Results on Hatefulmemes under various settings. "+" indicates the simultaneous application of two settings.

Better capturing task mapping consistently improves performance. As shown in Figure 2(a), across all shot counts, *EM* > *Standard* > *HM* in a clear descending order. This aligns with our observations from logit lens. In Figure 2(b), removing instructions, which serve as higher-level guidance enabling LVLMs to more deeply capture and utilize f_i, generally lowers performance. Yet "*EM* w/o $Inst$" still surpasses *Standard*.

Query sample is pivotal. Surprisingly, Figures 2(a) and (d) show that modifying $\hat{I}$ or $\hat{Q}$ causes greater performance variations than altering all ICDs. We hypothesize that LVLMs prioritize analyzing the query sample and use pretrained knowledge to constrain global task mapping accordingly.

Labels and visual modality matter, while task mapping takes precedence over them. While inverted labels degrade performance (Figure 2 (c)), clearer mappings compensate for these losses. Similarly, the performance decline caused by the lack of visual details can fully recover when task mappings are improved (Figure 2(d)). This suggests that both labels and visual modality affect multimodal ICL, but better utilization of task mapping can yield significant performance gains to address deficiencies in unimodal information.

2.3 ICL Needs Cohesive Task Mapping

Building on the central role of task mapping in multimodal ICL, we extend it to a more diversified scenario, termed **generalized-mapping tasks**. They reflect real-world ICL challenges where f_i exhibits nuanced or broad variability and specific-

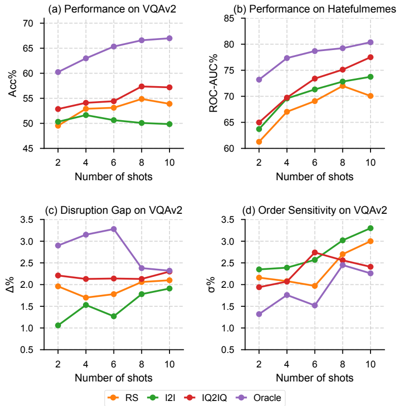


Figure 3: (a-b) Results of different ICL sequence configuration methods on VQAv2 and Hatefulmemes. (c-d) Task mapping cohesion analysis of different ICL sequence configuration methods on VQAv2.

mapping tasks can be viewed as a special case. They involve greater variability in Q_i and R_i, making component-level manipulations difficult. Thus, we turn to sequence-level configuration and demonstrate with an open-ended VQA dataset VQAv2 (Goyal et al., 2017).

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** 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 B.3). This greedy method goes beyond feature matching, though its reliance on $\hat{R}$ makes it impractical for real-world use.

Figure 3(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 ICL sequences that collectively resolve interdependent multimodal logic. Static methods like I2I, focused on isolated feature matching, disrupt cohesion and result in fragmented reasoning bias.

To validate this hypothesis, we evaluate task mapping cohesion using two metrics: Disruption Gap (Δ) and Order Sensitivity (σ) (details in Appendix B.5). 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 3(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.

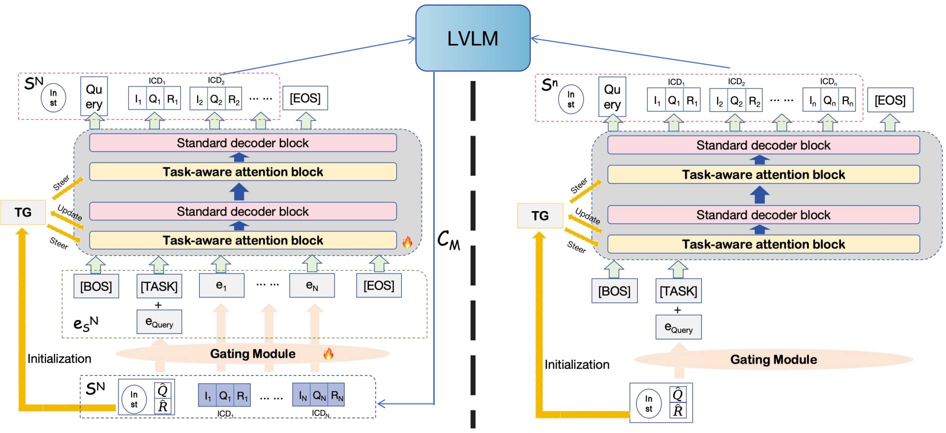
Finally, based on performance, Δ and σ, we identify four types of sequence, cases provided in Appendix B.6: (1)-(2) sequences impaired by isolated dependencies (e.g., similar image features and local task mapping bias), (3) sequences resembling specific-mapping tasks, and (4) the most common type, featuring diverse local mappings that collectively enhance cohesive task mapping. Such diversity enables LVLMs to overcome shallow reasoning and achieve superior multimodal ICL performance.

3 The Proposed Method

Section 2 highlights the critical role of task mapping. By modeling both local and global relationships within ICL sequences, task mapping ensures cohesive reasoning and task alignment. Motivated by these findings, we propose Task-aware model for ICL (Ta-ICL), a lightweight and end-to-end model designed to incorporate task mapping into sequence configuration and refinement.

Figure 4 illustrates the pipeline of Ta-ICL. The backbone of Ta-ICL consists of four transformer decoder blocks. It features a unique token vocabulary: instead of characters, each example (I_i, Q_i, R_i) from the given demonstration library DL is treated as an individual token. The training set D_S for Ta-ICL consists of N-shot ICL sequences. During training, Ta-ICL takes the entire N-shot ICL sequence as input. During inference, given a query sample and $Inst$, Ta-ICL can autoregressively retrieve n samples from DL to configure the optimal n-shot ICL sequence.

Input Embedding. Let x_i denote i-th ICD token (I_i, Q_i, R_i) and $\hat{x}$ denote the query sample $(\hat{I}, \hat{Q})$. In each input sequence, $\hat{x}$ is placed ahead of all x_i. To align with the autoregressive generation process, we use two special tokens, $[BOS]$



(a) Training

(b) Inference

Figure 4: Overview pipeline of Ta-ICL.

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 and concatenate it with $\hat{x}$ in the input sequence. It acts as a semantic anchor for task mapping. Therefore, for a given S^N, we reconstruct it as a token sequence $([BOS], [TASK]+\hat{x}, x_1, ..., x_N, [EOS]\})$. To filter and balance multimodal features for deeper mappings, we employ a binary gating module to generate the embedding e_i for x_i:

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

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

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:

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

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 $[TASK]$ with the embedding of $\hat{x}$ generated using the same gating module. The index of $\hat{e}$ is always 1 and I_{idx} denotes the index set of e_i.

$$M_{ij}^{(l)} = \begin{cases} \frac{\text{sim}(e_i, e_j)}{\sqrt{d}} \cdot \log\left(t_i^{(l)}\right), & j \leq i \text{ and } i, j \in I_{idx}, \\ \frac{\alpha\text{sim}(\hat{e}, e_j)}{\sqrt{d}} \cdot \log\left(t_1^{(l)}\right), & i = 1 \text{ and } j \in I_{idx}, \\ -\infty, & \text{otherwise.} \end{cases} \quad (7)$$

Task-aware Attention. The task-aware attention in Ta-ICL enables dynamic ICL 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 ICL sequences. TG encodes task intent through initialization by the multimodal fusion of the query sample and instruction:

$$e_{TG}^{(0)} = W_{TG} \cdot (E_I(\hat{I}) \oplus E_T(\hat{Q}) \oplus E_T(Inst')), \quad (8)$$

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 C.2).

Task-aware attention is applied selectively to certain layers, denoted as $\mathcal{L}_T$. At each of these layers, TG steers the attention mechanism by weighting relevance scores, which are derived from the interaction between TG and token embeddings. This interaction captures the hierarchical relationships between task mappings within the ICL sequence:

$$t_i^{(l)} = \sigma\left(\text{MLP}^{(l)}\left(e_{TG}^{(l)} \oplus e_i\right)\right), \quad (9)$$

where $\text{MLP}^{(l)}$: $\mathbb{R}^{2d} \rightarrow \mathbb{R}^d$ is a layer-specific network producing a scalar weight $t_i^l \in [0, 1]$ and σ is the sigmoid function. This weight reflects the degree to which each token contributes to the cohesive task mapping, dynamically adapting Ta-ICL's attention to emphasize semantically salient features. It modulates attention logits through a task-aware mask $M^{(l)}$. For intra-ICD tokens, the mask scales pairwise cosine similarities by $log(g_i^{(l)})$. For query-ICD tokens, a learnable coefficient α allows $\hat{e}$ to guide attention throughout the sequence. The mask is computed as follows for position (i, j): Here, the first case emphasizes interactions between local

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 ICL 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 9.

task mappings and the second case enable deep task mapping cohesion. The last case preserves the autoregressive nature. The mask $M^{(l)}$ is integrated into conventional attention, forming task-aware attention (TaAttn), as follows:

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

In particular, TG is updated between task-aware layers to preserve task mapping, 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 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 prompt a sharper representation of task-mapping. The total training objective combines the standard cross-entropy loss for sequence generation, sparsity regularization, and L2-norm constraint on TG to prevent overfitting:

Inference and Prompt Construction. After training, Ta-ICL can autoregressively select demonstrations from a library and configure ICL sequences. Given a new query sample $\hat{x}$, the input sequence to Ta-ICL during inference is $\{[BOS], [TASK] + \hat{x}\}$, where $\hat{x}$ is embedded using the trained gating module. The shot of 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 5. Ta-ICL then selects n ICDs using a beam search strategy with a beam size of 3, producing the optimal n-shot ICL 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 C.3.

4 Experiment

4.1 Training Data Construction and Models

We select six high-quality datasets across three key VL tasks to benchmark ICL 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 ICL 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

Configuration	VQA			Captioning		Classification	Hybrid	Fast	CLEVR
	VQAv2	VizWiz	OK-VQA	Flickr30K	MSCOCO	HatefulMemes			
Full Ta-ICL	**64.74**	**50.77**	**57.77**	**99.42**	**119.27**	**79.78**	**42.93**	**69.50**	**46.37**
(a) w/o [TASK] token	62.67	48.35	55.83	97.84	117.13	77.47	39.26	67.41	44.29
(b) w/o TG updates	60.18	47.54	54.47	97.51	116.92	75.63	36.80	65.38	42.81
(c) w/o $\mathcal{L}_{\text{sparse}}$	61.58	48.71	55.64	98.12	117.05	76.39	38.97	66.29	43.83
(d) w/o $\|W_{TG}\|_2^2$	58.14	46.15	53.95	97.73	118.28	73.34	34.95	66.31	42.96
(e) Random initialization	55.73	37.82	47.32	93.41	105.35	71.86	29.46	59.31	40.78
(f) w/o $\hat{I}$	61.39	47.21	54.68	96.52	114.73	76.26	37.62	66.38	43.51
(g) w/o $\hat{Q}$	59.46	46.07	54.05	95.78	112.61	74.32	35.87	65.49	42.35
(h) w/o $Inst'$	59.33	45.73	54.12	97.04	114.89	75.28	36.14	66.27	42.61

Table 2: Results of the ablation study on task mapping augmentation. Specifically, (a)-(d) correspond to diverse task-aware attention construction, (e)-(h) to diverse TG initialization.

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 2.3, 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 D.2. All S^N begin with a CoT-style $Inst$, as detailed in *Beginning1* of Table 3.

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 D.1.

4.2 Baselines and Implementation Details

We adopt RS and two similarity-based retrieval methods introduced in Section 2.3 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-$4n$ demonstrations based on joint similarity of I, Q, and R, and reranks 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 ICL 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 image and text encoders from CLIP-ViT-L/14 to generate image and text embeddings. For all 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.

4.3 Main Results

Table 1 summarizes the average ICL performance across five LVLMs under different ICL 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 construction of more precise cohesive task mappings. In Appendix D.4, we further investigate the impact of ICL sequence configuration on the LVLMs' multimodal ICL using per-model data.

5 Ablation Study

In this section, we focus on the impact of task-aware attention and the task mapping mechanism embedded within it.

Table 2 shows that each ablated component induces a complete performance degradation. TG, initialized by fusing the query's bimodal context with instruction semantics, establishes a task intent that aligns with the observation of Section 2: global mapping synthesis relies on query-driven grounding. Jointly anchored by the $[TASK]$ token, this intent prevents local mapping drift during autoregressive generation but also enables dynamic refinement through layered attention updates. By iteratively resolving coarse task boundaries into fine-

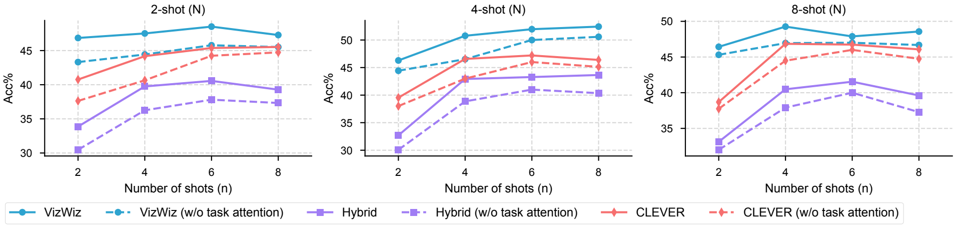


Figure 5: Results of Ta-ICL with and without task-aware attention under different N-n settings across three datasets, where N is the training sequence shot and n is the generation sequence shot.

grained patterns, TG harmonizes intra-sequence dependencies and query-context interactions, forming a feedback loop where each retrieved ICD sharpens global mapping cohesion. To conclude, task-aware attention effectively encodes task mapping as a dynamic attention-driven process, transcending static ICD aggregation to achieve consistent performance improvements in multimodal ICL.

To gain a deeper understanding of the role of task mapping throughout the entire process, we explore different combinations of training and generation shots. Our findings are as follows:

Task mapping consistently enhances multimodal ICL. Figure 5 shows that across all N-n combinations, task-aware attention always improves performance, highlighting the value of focusing ICL sequences on task mapping.

Cohesion remains robust as shots increase. For specific-mapping tasks (e.g., **CLEVR**), when N is fixed, performance gains diminish as n increases, while generalized-mapping tasks generally maintain steady improvements. This arises from each new ICD's unique contribution to the global task mapping, potentially deepening it rather than yielding diminishing returns on a specific mapping.

Task mapping enables flexibility in N and n. Although task-aware attention works best when N equals n, the cohesive design of task mapping allows Ta-ICL to effectively interpolate and extrapolate sequence shots across a flexible range of values. This adaptability ensures performance across diverse training data and enhances the model's potential for practical multimodal ICL applications, where flexibility and scalability are critical. We provide additional ablation studies in Appendix E.1, covering the construction of input embeddings, the role of instructions, the model's generalization to NLP and text-to-image tasks, and the cohesive task mappings in ICL sequences generated by Ta-ICL using the metrics in Section 2.3. The results show that training the encoder's last three layers and using binary gating enhance performance, while CoT-style instructions improve task alignment. These findings further validate the robustness and effectiveness of our task mapping framework.

6 Related Works

Interpreting In-Context Learning. The mechanisms of In-Context Learning (ICL) are crucial to better employing it (Gao et al., 2021; Dong et al., 2024). Min et al. (2022) attribute ICL's success to explicit information in ICDs like label space and input distribution, while Zhou et al. (2023) emphasize the importance of input-output mappings. To find a unified solution, Wei et al. (2023) and Pan et al. (2023) disentangle ICL into Task Recognition and Task Learning. Zhao et al. (2024) further propose a two-dimensional coordinate system to explain ICL behavior via two orthogonal variables: similarity in ICDs and LLMs' ability to recognize tasks. However, these studies are often confined to tasks with small label spaces and struggle to address complex multimodal scenarios.

7 Conclusion

This work introduces a novel perspective on multimodal ICL by focusing on task mapping. We systematically demonstrate the principles and critical importance of task mapping within ICL sequences for enabling efficient ICL in LVLMs. These insights further inspire leveraging task mapping to enhance ICL sequence configuration. To this end, we propose Ta-ICL, which employs task-aware attention to deeply integrate task mapping into the autoregressive process, thereby optimizing sequence configuration. Experiments show consistent outperformance over SOTA baselines, particularly in generalized-mapping tasks. This study not only presents a practical model but also provides the multimodal ICL community with a new and reliable research direction.

Limitations

Despite its contributions, this work has certain limitations. First, while we emphasize the importance of task mapping in ICL, we do not establish a formal mathematical framework to define or model task mapping. Such a framework could provide more rigorous insights into the construction and evaluation of task mapping and will be a valuable direction for future research.

Second, this study does not delve into the role of LVLMs' internal attention mechanisms and hidden state in capturing and utilizing task mapping. Investigating how task mapping manifests within attention layers could uncover deeper connections between sequence configuration and model reasoning, offering another promising avenue for out future work.

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. 2022. Flamingo: a visual language model for few-shot learning. *Preprint*, arXiv:2204.14198.

Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. 2023. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*.

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. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Wentong Chen, Yankai Lin, ZhenHao Zhou, HongYun Huang, Yantao Jia, Zhao Cao, and Ji-Rong Wen. 2024a. Icleval: Evaluating in-context learning ability of large language models. *Preprint*, arXiv:2406.14955.

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

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. *Preprint*, arXiv:2301.00234.

Caoyun Fan, Jidong Tian, Yitian Li, Hao He, and Yaohui Jin. 2024. Comparable demonstrations are important in in-context learning: A novel perspective on demonstration selection. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10436–10440.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. *Preprint*, arXiv:2012.15723.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. 2023. What can transformers learn in-context? a case study of simple function classes. *Preprint*, arXiv:2208.01066.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. 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*, pages 6904–6913.

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

Dan Iter, Reid Pryzant, Ruochen Xu, Shuohang Wang, Yang Liu, Yichong Xu, and Chenguang Zhu. 2023. In-context demonstration selection with cross entropy difference. *Preprint*, arXiv:2305.14726.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. *Advances in neural information processing systems*, 33:2611–2624.

Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander M. Rush, Douwe Kiela, Matthieu Cord, and Victor Sanh. 2023. Obelics: An open web-scale filtered dataset of interleaved image-text documents. *Preprint*, arXiv:2306.16527.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *Preprint*, arXiv:2104.08691.

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

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

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021a. What makes good in-context examples for gpt-3? *Preprint*, arXiv:2101.06804.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021b. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Preprint*, arXiv:2107.13586.

Xinxi Lyu, Sewon Min, Iz Beltagy, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Z-icl: Zero-shot in-context learning with pseudo-demonstrations. *Preprint*, arXiv:2212.09865.

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

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *Preprint*, arXiv:2202.12837.

nostalgebraist. 2020. interpreting gpt: the logit lens. *LessWrong*.

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. 2022. In-context learning and induction heads. *Preprint*, arXiv: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. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Jane Pan, Tianyu Gao, Howard Chen, and Danqi Chen. 2023. What in-context learning "learns" in-context: Disentangling task recognition and task learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8298–8319, Toronto, Canada. Association for Computational Linguistics.

Vinay M. S., Minh-Hao Van, and Xintao Wu. 2024. In-context learning demonstration selection via influence analysis. *Preprint*, arXiv:2402.11750.

Sarah Schwettmann, Neil Chowdhury, Samuel Klein, David Bau, and Antonio Torralba. 2023. Multimodal neurons in pretrained text-only transformers. In *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 2854–2859.

Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. 2024. Emu: Generative pretraining in multimodality. *Preprint*, arXiv:2307.05222.

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

Liang Wang, Nan Yang, and Furu Wei. 2024. Learning to retrieve in-context examples for large language models. *Preprint*, arXiv:2307.07164.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. 2023. Larger language models do in-context learning differently. *Preprint*, arXiv:2303.03846.

Yang Wu, Shilong Wang, Hao Yang, Tian Zheng, Hongbo Zhang, Yanyan Zhao, and Bing Qin. 2023a. An early evaluation of gpt-4v(ision). *Preprint*, arXiv:2310.16534.

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2023b. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. *Preprint*, arXiv:2212.10375.

Xu Yang, Yingzhe Peng, Haoxuan Ma, Shuo Xu, Chi Zhang, Yucheng Han, and Hanwang Zhang. 2024. Lever lm: Configuring in-context sequence to lever large vision language models. *Preprint*, arXiv:2312.10104.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. 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.

Yu Yuan, Lili Zhao, Kai Zhang, Guangting Zheng, and Qi Liu. 2024. Do llms overcome shortcut learning? an evaluation of shortcut challenges in large language models. *Preprint*, arXiv:2410.13343.

Anhao Zhao, Fanghua Ye, Jinlan Fu, and Xiaoyu Shen. 2024. Unveiling in-context learning: A coordinate system to understand its working mechanism. *Preprint*, arXiv:2407.17011.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. *Preprint*, arXiv:2205.10625.

Yucheng Zhou, Xiang Li, Qianning Wang, and Jianbing Shen. 2024. Visual in-context learning for large vision-language models. *Preprint*, arXiv:2402.11574.

Yongshuo Zong, Ondrej Bohdal, and Timothy Hospedales. 2024. Vl-icl bench: The devil in the details of multimodal in-context learning. *Preprint*, arXiv:2403.13164.

A Related Works

Configuring ICD sequences. Similarity-based retrieval fails to provide LLMs with the deep task mappings (Liu et al., 2021a; Li et al., 2024). The ICD bias introduced by coarse-grained retrieval also amplifies the short-cut effect (Lyu et al., 2023; Yuan et al., 2024). Model-dependent methods have also emerged later, employing multiple models for more demanding selection (Wu et al., 2023b; Wang et al., 2024; S. et al., 2024). These methods are not end-to-end and increasing overly focus on ICD selection over ordering. One work closely connected to ours is Yang et al. (2024), which introduces a tiny language model composed of two encoder blocks to automatically select and order ICDs. However, it is limited in complex tasks without a deep insight of task mapping.

B Viison-lanuage In-context Learning

B.1 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 "Describe the whole image in a short sentence. " 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.

(d) **Fast Open-ended MiniImageNet**: Both Q_i and $\hat{Q}$ are set as short prompts instructing the LVLM to recognize the object in image, such as "This is an image of:" The result R_i is the self-defined label.

(e) **CLEVR Counting Induction**: Both Q_i and $\hat{Q}$ are implicit texts in the form of "attribute: value" pairs. The result R_i is the number of objects matching the pairs.

For all the tasks mentioned above, since the ground-truth answers are not visible to the LVLM during reasoning, all $\hat{R}$ are set to blank. The visualization of (I, Q, R) triplets for the four tasks is shown in Figure 6.

B.2 Logit Lens

"Shallow" represents superficial task understanding, focusing on general or surface-level concepts. Anchor words include "category," "judge," "label," "identify," and "predict." "Deep" indicates a more profound comprehension of the task, capturing nuanced or context-sensitive meanings. Anchor words include "hateful," "offensive," "biased," "harmful," and "inappropriate." "Correct" corresponds to the correct answer for the query sample. "Wrong" represents the incorrect answer, opposite to "Correct."

B.3 Oracle

Oracle uses the same LVLM $\mathcal{M}$ for both configuring the ICL sequences and performing ICL. This method aims to construct high-quality ICL 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:

$$\mathcal{C}_{\mathcal{M}}(S^n) = \sum_t logP_{\mathcal{M}}(\hat{R}^{(t)} \mid S^n, \hat{R}^{(1:t-1)}), \tag{13}$$

where $\mathcal{M}$ denotes the LVLM. This score measures how effectively the model predicts the ground-truth result $\hat{R}$ given the current ICL 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:

$$x_n = \underset{x \in D}{argmax}[\mathcal{C}_{\mathcal{M}}(S^{n-1} + x) - \mathcal{C}_{\mathcal{M}}(S^{n-1})]. \tag{14}$$

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.

B.4 Ablation Settings

To systematically evaluate the impact of task mapping in multimodal in-context learning (ICL), we design controlled ablation settings that selectively perturb key factors such as **label reliability** and **visual modality**. Below, we provide detailed descriptions of each setting's implementation.

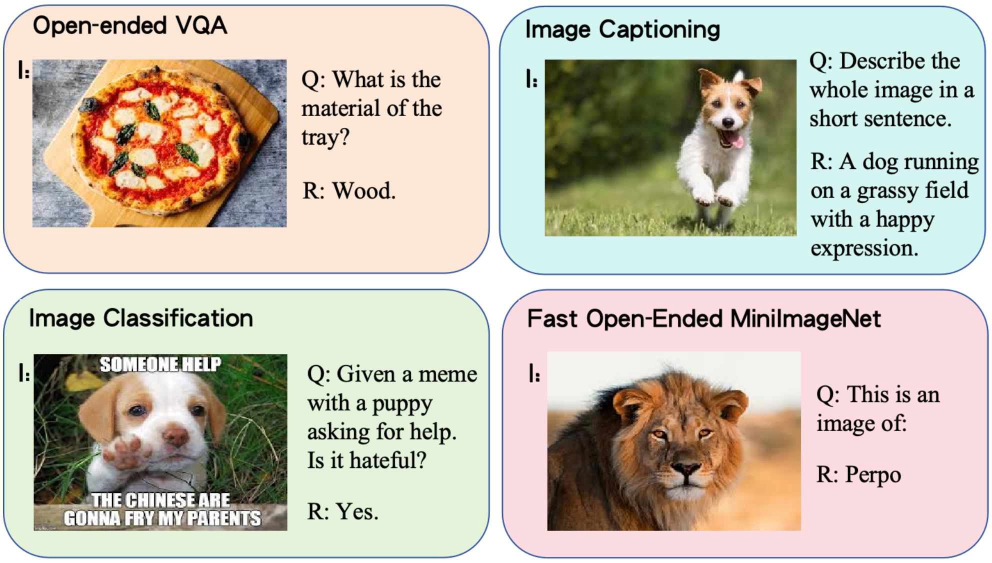

Open-ended VQA
I:
Q: What is the material of the tray?
R: Wood.
Image Captioning
I:
Q: Describe the whole image in a short sentence.
R: A dog running on a grassy field with a happy expression.
Image Classification
I:
SOMEONE HELP
THE CHINESE ARE GONNA FRY MY PARENTS
Q: Given a meme with a puppy asking for help. Is it hateful?
R: Yes.
Fast Open-Ended MiniImageNet
I:
Q: This is an image of:
R: Perpo

Figure 6: The visualization of (I, Q, R) triplets for Open-ended VQA, image captioning, image classification and Fast Open-ended MiniImageNet.

1. **Label Reliability**
   - *Wrong Labels (WL)*: To evaluate the reliance on explicit label correctness, we invert 75% R_i labels (yes↔no) in the ICL sequence. This setting disrupts direct label-based learning while maintaining the overall task structure, allowing us to examine whether LVLMs primarily depend on task mapping rather than correct labels.
2. **Visual modality**
   - *Blur Images (BI)*: To investigate the role of visual information clarity, we apply Gaussian blur to the images I_i in the ICL sequence. This degrades fine-grained details while preserving overall structure, allowing us to examine the impact of visual degradation on task mapping.
   - *BI on Query Image (BI($\hat{I}$))*: Instead of applying blur to the entire ICL sequence, (BI)$\hat{I}$ applies Gaussian blur only to the query image $\hat{I}$. This setting helps isolate the effect of degraded query information on task mapping performance.
3. **Query Enhancement**
   - *Easier Mapping on Query (EM($\hat{Q}$))*: This setting enhances the query text $\hat{Q}$ by incorporating explicit task guidance to facilitate task mapping. Instead of modifying the ICL sequence, EM($\hat{Q}$) provides additional textual hints that reinforce task semantics, allowing us to measure whether improved query understanding compensates for suboptimal ICD configurations.

B.5 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.

B.5.1 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:

$$\Delta_i = \left| \mathcal{L}(S) - \mathcal{L}(\mathcal{S}_{\text{replaced},i}) \right|, \tag{15}$$

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:

$$\Delta = \frac{1}{N} \sum_{i=1}^{N} \Delta_i. \tag{16}$$

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.

B.5.2 Order Sensitivity (σ)

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

$$S^n_{\text{permute},1}, S^n_{\text{permute},2}, \ldots, S^n_{\text{permute},K}, \quad K = 10. \tag{17}$$

Then we compute the accuracy for each permuted sequence:

$$\text{Acc}(S^n_{\text{permute},k}) = \frac{\text{Correct Predictions}}{\text{Total Predictions}}, \quad k = 1, 2, \ldots, K. \tag{18}$$

Then calculate the mean accuracy across all permutations:

$$\mu = \frac{1}{K} \sum_{k=1}^{K} \text{Acc}(S^n_{\text{permute},k}). \tag{19}$$

Finaly, compute the standard deviation of accuracies as σ:

$$\sigma = \sqrt{\frac{1}{K} \sum_{k=1}^{K} \left(\text{Acc}(S^n_{\text{permute},k}) - \mu \right)^2}. \tag{20}$$

B.6 Case Study

In Figure 7, we present four examples representing the four typical types of ICL sequences in generalized-mapping tasks.

C Method

C.1 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.

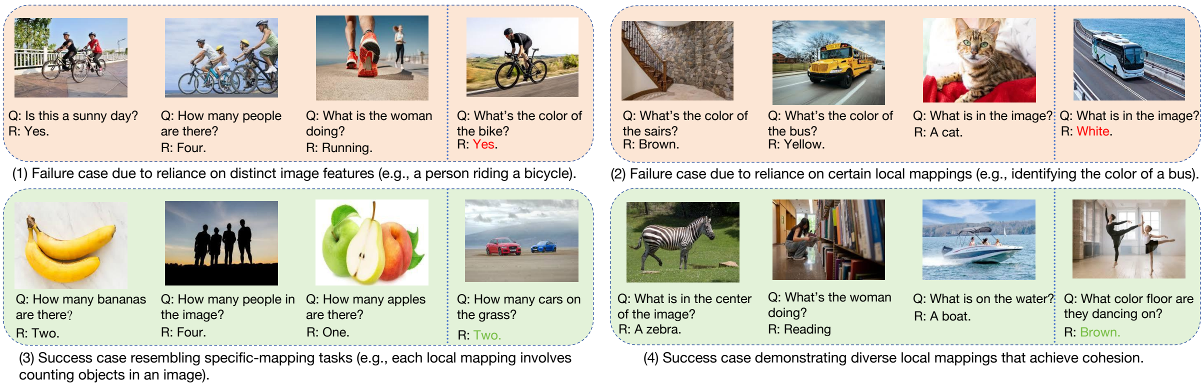
CLIP is trained to align the embedding spaces of images and text through a contrastive learning objective. Specifically, the model optimizes a contrastive loss that increases the cosine similarity for matched image-text pairs, while reducing it for unmatched pairs within each training batch. To ensure 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.

C.2 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 ICL 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


Q: Is this a sunny day? R: Yes.
Q: How many people are there? R: Four.
Q: What is the woman doing? R: Running.
Q: What's the color of the bike? R: Yes.
Q: What's the color of the sairs? R: Brown.
Q: What's the color of the bus? R: Yellow.
Q: What is in the image? R: A cat.
Q: What is in the image? R: White.
Q: How many bananas are there? R: Two.
Q: How many people in the image? R: Four.
Q: How many apples are there? R: One.
Q: How many cars on the grass? R: Two.
Q: What is in the center of the image? R: A zebra.
Q: What's the woman doing? R: Reading
Q: What is on the water? R: A boat.
Q: What color floor are they dancing on? R: Brown.

(1) Failure case due to reliance on distinct image features (e.g., a person riding a bicycle).

(2) Failure case due to reliance on certain local mappings (e.g., identifying the color of a bus).

(3) Success case resembling specific-mapping tasks (e.g., each local mapping involves counting objects in an image).

(4) Success case demonstrating diverse local mappings that achieve cohesion.

Figure 7: Four types of ICL sequences in generalized-mapping task.

issues caused by too long instructions.

As shown in Table 3, 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.

C.3 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 "Query:" and its result starts with "Result:". The query sample concludes with "Result:", 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 4 provides an overview of the prompt details used for the different models in our experiments. Each model, including OpenFlamingoV2, ICDEFICSv1, 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 predictions in VL tasks.

D Experiment

D.1 Datasets and Models

D.1.1 Dataset

In our study, we explore various VL tasks that use diverse datasets to evaluate model performance. As illustrated in Figure 8, 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 HatefulMemes 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

$Inst$	**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 generate the result of the given query.
Beginning2 (PPT)	Construct a dynamic pattern repository from image-text samples, then leverage this framework alongside your knowledge base for concurrent visual analysis and query 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 (CoT)	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 generate the result of the given question.
End (CoT)	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 generate the result of the given question.
Beginning1 (Abbreviated)	Analyze the following image-text pairs, understand the task, and use this to generate the result with a new image.
Middle (Abbreviated)	After reviewing the above image-text pairs, analyze the task and use this understanding to generate the result with a new image.
End (Abbreviated)	After reviewing the above image-text pairs and a query with a new image, analyze the task and use this understanding to generate the result.

Table 3: 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	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 generate the result of the given query. <img><IMG_CONTEXT><|endofchunk|> Query: In what country can you see this? Result: vietnam <img><IMG_CONTEXT><|endofchunk|> Query: Is this a buggy or car? Result: buggy <img><IMG_CONTEXT><|endofchunk|> Query: What is this? Result:
IDEFICS2	"User: 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 generate the result of the given query." "\nUser:<|image_pad|> Query: In what country can you see this? <end_of_utterance>", "\nAssistant: Result: vietnam. <end_of_utterance>", "\nUser: <|image_pad|> Query: Is this a buggy or car? <end_of_utterance>", "\nAssistant: Result: buggy. <end_of_utterance>", <|image_pad|> Query: What is this? <end_of_utterance>", "\nAssistant: Result:"
InternVL2	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 generate the result of the given query. <img><IMG_CONTEXT></img> Query: In what country can you see this? Result: vietnam <img><IMG_CONTEXT></img> Query: Is this a buggy or car? Result: buggy <img><IMG_CONTEXT></img> Query: What is this? Result:
Qwen2VL	<|im_start|>system You are a helpful assistant.<|im_end|> <|im_start|>user 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 generate the result of the given query. <|vision_start|><|image_pad|><|vision_end|>Query:In what country can you see this? Result: vietnam <|vision_start|><|image_pad|><|vision_end|>Query: Is this a buggy or car? Result: buggy <|vision_start|><|image_pad|><|vision_end|>Query: What is this? Result: <|im_end|> <|im_start|>assistant

Table 4: 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.

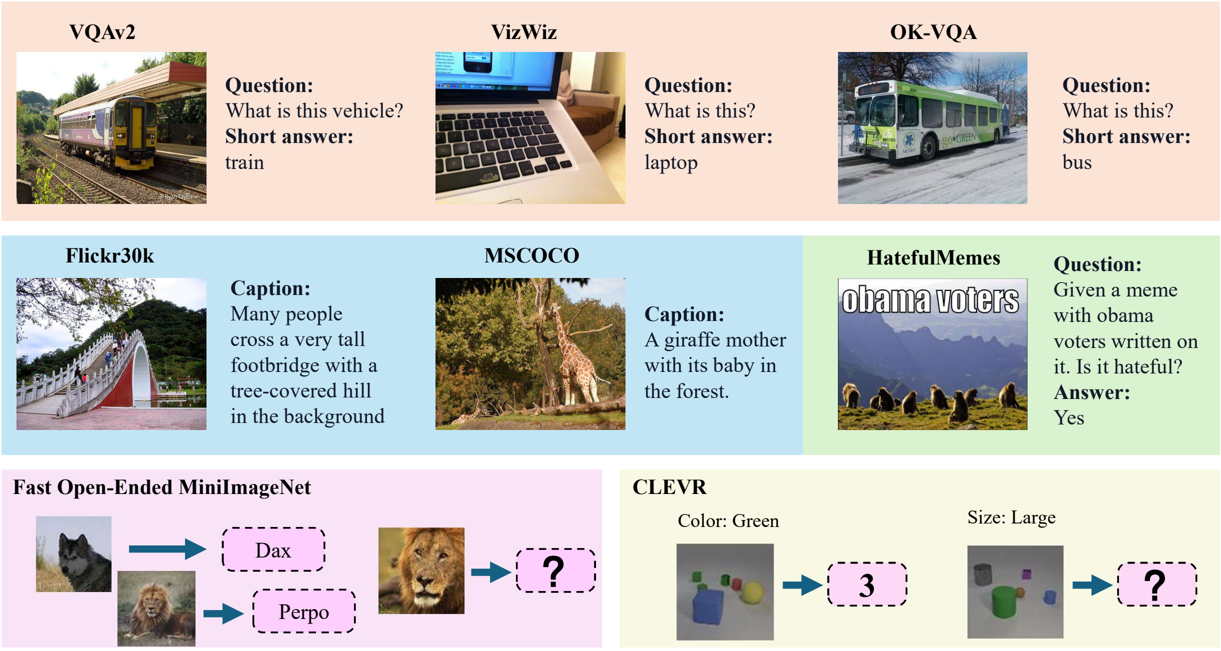


Figure 8: Illustrative examples from various vision-and-language datasets categorized by task type. Visual Question Answering (VQA) tasks are shown in red (VQAv2: train, VizWiz: laptop, OK-VQA: bus). Captioning tasks are represented in blue (Flickr30k: footbridge, MSCOCO: giraffes), while classification 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).

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.

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 pattern 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 6. 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:

$$Acc_{a_i} = max(1, \frac{3 \times \sum_{k\in[0,9]} match(a_i, g_k)}{10}), \tag{21}$$

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.

D.1.2 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 <|vision_start|> and <|vision_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.

D.2 Training Data Construction Details

We construct sequence data for model training using existing high-quality datasets, each corresponding to a VL task. The samples are uniformly formatted as (I, Q, R) triplets based on their respective task types. Each dataset generates a sequence set D_S for training, where each sequence consists of a query sample and N ICDs. The value of N is configurable, determining the number of shots during training. To ensure optimal training performance,

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

Table 6: 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.

we employ the same LVLM used in inference as a scorer to supervise the construction of D_S, making the method inherently model-specific. For each dataset, we construct D_S exclusively from its training set through the following three-step process: (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:

$$\mathcal{C}_{\mathcal{M}}(S_i^n) = \sum_t logP_{\mathcal{M}}(\hat{R}_i^{(t)} \mid S_i^n, \hat{R}_i^{(1:t-1)}), \tag{22}$$

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 $\mathcal{C}_{\mathcal{M}}$:

$$x_n = \underset{x \in D_i}{argmax}[\mathcal{S}\rfloor_{\mathcal{M}}(S_i^{n-1} + x) - \mathcal{S}\rfloor_{\mathcal{M}}(S_i^{n-1})]. \tag{23}$$

(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.

D.3 Baselines

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 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 ICL sequence configuration. Observing that configuring an ICL sequence resembles composing a sentence, Lever-LM leverages a temporal learning approach to identify these patterns. A special dataset of effective ICL 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 ICL sequence configuration for LVLMs.

Datasets	**Training**	**Validation**	**Test**	$\hat{D}$ **Size**	metrics
Rule Learning	1600	-	150		exact match scores
Fast Counting	800	-	40		Accuracy

Table 7: Overview of Rule Learning and Fast Counting tasks.

$Inst'$	$Inst$	VQAv2	Vi2Wi2	OKVQA	**Hybrid**
Beginning1	*Beginning2*	59.48	47.28	54.63	34.92
	Beginning3	57.40	45.26	52.10	30.21
	End	63.24	50.69	55.19	30.37
Beginning2	*Beginning1*	64.17	49.27	56.12	37.60
Beginning3		63.78	49.36	55.43	35.98
End		63.62	49.08	55.68	36.43

Table 8: Results of Ta-ICL under various $Inst'$-$Inst$ combinations. $Inst'$ represents the style used for initializing TG, while $Inst$ refers to the style actually incorporated into the prompt.

D.4 Results and Analysis

We can go deep into the per-model results in Tabel 9. 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., 2023a). This results in the quality improvement of the ICL 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 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 ICL sequence quality.

E Ablation Study

E.1 Input Embeddings

Input Embedding. To investigate the impact of input embedding construction on ICL sequence configuration, we vary both the training method of the CLIP encoders and the adoption of the gating module to evaluate Ta-ICL's performance under different settings. For the CLIP encoders, we explore three alternative methods: one involves freezing its parameters and adding an MLP adapter to its output, which is then trained; another involves fully training the entire encoder; and the third involves training only the last two layers. For constructing the embeddings multimodal ICD tokens, we first experimented with direct concatenation without gating modules:

$$e_i = E_I(I_i) + E_T(Q)_i + E_T(R_i) + r_i, \quad (24)$$

where r_i is a randomly initialized learnable component introduced into the embedding. Besides binary gating, we examine a finer-grained ternary gating module that assigns separate weights to control the contributions of all three components I, Q and R:

$$e_i = g_I \cdot E_I(I_i) + g_Q \cdot E_T(Q_i) + g_R \cdot E_T(R_i), \quad (25)$$

where g_I, g_Q and g_R denote the weights computed using a softmax function applied the linear transformations, ensuring their sum equals 1. Additionally, we apply regularization to the weights: $g_I^2 + g_Q^2 + g_R^2 \leq \theta$ to prevent excessive reliance on specific components.

The training approach for CLIP affects the feature representation of embeddings, which in turn influences Ta-ICL's ability to capture cross-modal details during sequence configuration. From Table 10 we observe that for tasks with intrinsic features like VQA and **Hybrid**, leaving the CLIP unchanged or only adding an adapter leads to significant degradation in the quality of the ICL sequence generation. In fact, even methods that only train the last two layers show a more noticeable performance gap compared to the current approach. This highlights that the output pattern of the third-to-last layer of the encoder is crucial for capturing core task features in multimodal ICD. When we replaced our current training method with one that fully trains CLIP, we did not observe a significant performance drop. This suggests that Ta-ICL's treatment of ICDs as tokens does not cause feature loss. In contrast, through task-aware attention, it enhances feature representation, helping mitigate the limitations of the embedding itself. Considering the high cost of training the entire encoders, current method is optimal.

As we point out in Section 2, it is important for the model to focus on fine-grained features within

		VQA			**Captioning**		**Classification**	**Hybrid**	**Fast**	**CLEVR**
		VQAv2	VizWiz	OK-VQA	Flickr30K	MSCOCO	HatefulMemes			
OpenFlamingov2	RS	50.84	27.71	37.90	76.74	92.98	64.75	13.48	57.69	21.60
	I2I	49.52	26.82	37.79	79.84	94.31	69.53	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	**61.12**	**39.76**	**47.28**	**84.23**	**99.10**	**75.09**	**35.17**	**60.25**	**24.80**
IDEFICS2	RS	54.97	32.92	40.01	82.43	99.61	69.31	15.65	54.72	35.14
	I2I	53.77	31.67	41.37	85.76	101.34	69.64	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	**59.41**	**38.32**	**48.35**	**90.41**	**107.04**	**73.68**	**33.25**	**57.21**	**40.21**
InternVL2	RS	63.35	54.70	57.13	99.05	116.37	70.72	17.74	75.87	57.03
	I2I	61.83	55.07	58.73	103.29	118.46	76.27	14.82	75.89	54.79
	IQ2IQ	64.57	56.94	**63.91**	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	**69.42**	**61.69**	63.27	**108.26**	**128.34**	**82.97**	**45.79**	**80.76**	**58.27**
Qwen2VL	RS	64.28	48.97	55.30	100.32	121.47	77.85	20.42	66.29	48.70
	I2I	63.71	48.75	56.39	102.87	124.50	80.62	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	**72.87**	**57.93**	**64.97**	**106.91**	**132.14**	**83.19**	**48.95**	**73.09**	**53.98**
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**	**68.89**	**107.29**	**129.71**	83.96	**51.48**	**64.17**	**50.59**

Table 9: 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.

Ta	VQAv2	MSCOCO	Hatefulmemes	**Hybrid**	**Fast**	**CLEVR**
(CLIP Encoder)						
N/A	20.41	98.26	47.82	14.80	48.67	20.52
Adapter only	25.37	108.54	67.85	18.93	54.29	25.71
Fully training	**47.57**	114.46	**76.29**	**37.43**	63.49	**43.22**
Last two	42.63	114.25	73.18	28.91	62.13	39.27
Last three	46.81	**114.79**	75.60	35.91	**63.72**	42.18
(Gating Module)						
+ Ternary gating	47.21	113.92	**80.02**	37.64	65.48	44.89
+ Binary gating	**50.77**	**119.27**	79.78	**42.93**	**69.50**	**46.57**

Table 10: Results of Ta-ICL with different input embedding configurations. (CLIP Encoder) section shows the results without adding gating modules under various training methods for CLIP encoders. N/A indicates no training or modification. (Gating Module) section presents the results with two gating modules added on top of the encoders trained with the method of training the last three layers.

the two modalities for multimodal ICL. However, Table 10 shows that the use of a ternary gating mechanism to obtain more refined embeddings actually results in worse performance compared to binary gating, likely due to insufficient parameter capacity in Ta-ICL.

E.2 Instruction

Sections 2.2 and 5 highlight the importance of $Inst$ in improving multimodal ICL performance. However, as shown in Table 12, using the original embedding of $Inst$ to initialize TG degrades Ta-ICL performance due to semantic redundancy from long text embeddings, which can cause TG deviation and hinder convergence.

We further investigated the effects of $Inst$'s style and position. Two new styles were developed and placed at the beginning of the prompt, while the CoT-style was also tested between the ICDs and query sample, as well as at the end. Details are provided in Appendix C.2. Table 12 shows that $Inst$'s position has minimal impact, but its style significantly affects performance, with the CoT-style being the most effective. Moreover, as discussed in Appendix E.2, Ta-ICL demonstrates limited sensitivity to style changes, with the style's influence primarily arising from LVLMs. Thus, $Inst$ can be viewed as a special ICD, contributing high-level local task mapping that integrates into the LVLM's global task mapping. Table 8 shows that when the instruction used for TG initialization and the one included in the prompt have different styles, Ta-ICL demonstrates greater robustness. Changes in the style of $Inst'$ not only result in minimal performance degradation but also lead to significantly smaller performance variations. In contrast, for LVLMs, changes in $Inst$ style cause noticeable performance gaps and a clear preference for specific styles. This indicates that the performance fluctuations caused by $Inst$ are primarily attributable to LVLMs rather than Ta-ICL itself.

E.3 Generalization Test

To demonstrate the generalization of Ta-ICL beyond image-to-text tasks, we evaluate its performance on NLP and text-to-image tasks. We first use the latest LLM ICL benchmark, ICLEval's (Chen et al., 2024a) Rule Learning part to construct a mixed-task NLP dataset and test it on Qwen-7B and LLaMA3-8B. For text-to-image tasks, we use the Fast Counting dataset from the VL-ICL Bench and test it on Emu2-Gen (Sun et al., 2024). The ICDs in both tasks can be represented as (Q, R). Results in Table 13 show that Ta-ICL consistently outperforms baselines across all tasks, highlighting its strong generalizability and wide application potential.

For NLP evaluation, we utilize the Rule Learning part of the latest benchmark, ICLEval. ICLEval is designed to assess the ICL abilities of LLMs, focusing on two main sub-abilities: exact copying and rule learning. The Rule Learning part evaluates how well LLMs can derive and apply rules from examples in the context. This includes tasks such as format learning, where models must replicate and adapt formats from given examples, and order and statistics-based rule learning, where the model must discern and implement patterns such as item sequencing or handling duplications. These tasks challenge LLMs to go beyond language fluency, testing their ability to generalize from context in diverse scenarios. Examples of (Q, R) pairs can be found in Table 14. For all tasks, we use exact match scores to evaluate the predictions with the labels.

For text-to-image evaluation, we utilize the Fast Counting task in the VL-ICL bench. In this task, artificial names are associated with the counts of objects in the image. The task is to generate an image that shows a given object in quantity associated with the keyword (e.g. perpo dogs where perpo means two). Thus, each Q is a two-word phrase such as 'perpo dogs', and its corresponding R is an image of two dogs.

The ICDs in both tasks can be represented as (Q, R). In NLP, both Q and R are text; in text-to-image, Q is text while R is an image. We simply need to adjust the embedding encoder and gating module accordingly. The baselines are RS, **Q2Q**

Experiment	**Type**	**Position**	VizWiz	MSCOCO	Hatefulmemes	**Hybrid**	**Fast**	**CLEVR**
	None	N/A						

Table 11: Results for different instruction types and positions across various datasets.

Instruction	VizWiz	MSCOCO	Hatefulmemes	**Hybrid**	**Fast**	**CLEVR**
Beginning1	**50.77**	119.27	**79.78**	42.93	**67.10**	**45.57**
$Inst' \rightarrow Inst$	42.89	113.81	75.62	24.97	63.78	37.69
Beginning2	48.98	118.65	78.13	41.07	66.40	44.86
Beginning3	47.30	117.26	77.88	40.01	65.67	43.51
Middle	50.13	**119.83**	79.53	42.64	66.85	43.91
End	50.24	118.69	79.60	**43.18**	66.71	45.08

Table 12: Results of Ta-ICL with diverse instruction types. The highest scores are highlighted in **bold**. $Inst' \rightarrow Inst$ means using $Inst$ during the initialization of TG.

Methods	**NLP**		**text-to-image**
	Qwen-7B	LLaMA3-8B	Emu2-Gen
RS	0.26	0.30	43.67
Q2Q	0.46	0.54	47.83
QPR	0.45	0.56	49.06
Lever-LM	0.47	0.60	-
Ours	**0.50**	**0.61**	**51.18**

Table 13: Results of different ICL sequence configuration methods in NLP and text-to-image tasks. Both training and generated shots are set to 4. The highest scores are highlighted in **bold**.

(Query-to-query), **QPR** (Query&pseudo-result), and Lever-LM (not applicable to text-to-image).

E.4 Task Mapping Cohesion

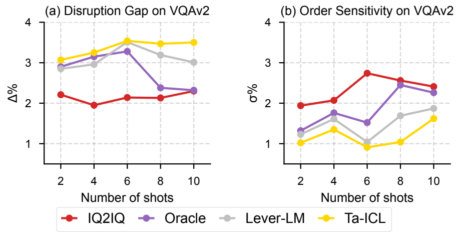


Figure 9: Analysis of task mapping cohesion in n-shot ICL sequences generated by different methods.

We again utilize the two metrics introduced in Section 2.3, Disruption Gap (Δ) and Order Sensitivity (σ), to evaluate task mapping cohesion in ICL sequences generated by Ta-ICL. Figure 9 shows that Ta-ICL achieves the highest Δ and lowest σ across all shots. This not only indicates that Ta-ICL-generated ICL sequences construct robust task mappings effectively utilized by LVLMs but also provides further evidence supporting the validity of our task mapping framework. 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.

Task	Q	R
Format rules	|Index|name|age|city| |—|—|—|—| |1|Elijah Morgan|36|Pittsburgh|	<person> <name>Elijah Morgan</name> <age>36</age> <city>Pittsburgh</city> </person>
Statistics rules	588 and 823 are friends. 885 and 823 are friends. 795 and 588 are friends. 890 and 823 are friends. 885 and 588 are friends. 890 and 588 are friends. 795 and 823 are friends. Query: Who are the friends of 885?	823, 588
Order rules	Input: activity, brief, wonder, anger Output: anger, wonder, activity, brief Input: market, forever, will, curve Output: curve, will, market, forever Input: pain, leading, drag, shoot Output: shoot, drag, pain, leading Input: shopping, drama, care, start Output:	start, care, shopping, drama
List Mapping	Input: [1, 3, 6, 1, 83] Output: [3] Input: [5, 6, 35, 3, 67, 41, 27, 82] Output: [6, 35, 3, 67, 41] Input: [8, 45, 6, 18, 94, 0, 1, 2, 7, 34] Output: [45, 6, 18, 94, 0, 1, 2, 7] Input: [2, 7, 66, 6, 93, 4, 47] Output:	[7, 66]

Table 14: The examples of four Rule Learning tasks in ICLEval.