EXPRESSION MATTERS: TASK ADAPTATION THROUGH SELF-GENERATED SUPERVISION

Anonymous authors
Paper under double-blind review

ABSTRACT

While the next-token prediction (NTP) paradigm enables large language models (LLMs) to be versatile for various tasks, the NTP interface is not naturally matched to non-generative tasks whose desired outputs are not token sequences. We argue that LLM performance can be limited not only by what models know, but also by how they express what they know. We refer to the latter as the *knowledge expression bottleneck*. We introduce Self-Knowledge Re-expression (SKR), a task-general method that relies on a model's self-generated signals to align its output mechanism with a task-native format. SKR requires only unlabeled task inputs and an explicit task configuration, without relying on any human-provided annotations or external teacher models' outputs. This makes SKR much more cost-efficient to implement and allows it to be executed locally without sending task data to any external annotation service. Experiments across multiple tasks on an industrial financial-document dataset demonstrate substantial benefits of SKR. It achieves an absolute Recall@1 gain of at least 40% on information retrieval, reduces latency by at least 76% for instruction-conditioned region localization, and achieves an absolute AUPRC gain of at least 33% on imbalanced binary classification. On MMDocRAG, the strongest SKR model exceeds the best published baseline by an absolute 16.9% in Recall@10. These results show that, in addition to knowledge acquisition, knowledge expression is an important bottleneck in LLM task performance and that SKR provides a cost-efficient method for task adaptation.

1 INTRODUCTION

Pre-training enables LLMs to acquire broad linguistic and reasoning capabilities, and scaling model size and training data has been a central route to expanding this knowledge (Radford et al., 2018; 2019; Brown et al., 2020; Kaplan et al., 2020; Hoffmann et al., 2022). However, task performance depends not only on what a model knows, but also on how it converts task-relevant information into the required outputs. We call this process *knowledge expression*. We argue that knowledge expression is a critical, yet under-explored, factor in LLMs' task performance.

Most contemporary LLMs are trained with next-token prediction (NTP) and rely on this token generation interface to execute tasks. This interface is effective for language generation, but discrete, sequential strings are not a natural output space for every task. For example, as illustrated in Figure 1, retrieval requires representations organized by semantic proximity; localization requires continuous coordinates; and imbalanced classification benefits from class scores rather than vocabulary distributions. In such cases, task performance can be constrained by the output mismatch even when the base model's behavior already contains useful task-relevant signals.

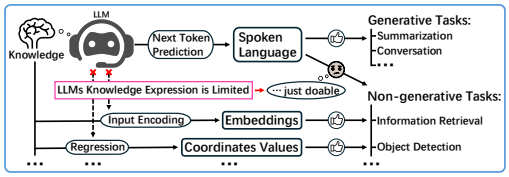


Figure 1: An autoregressive token interface is broadly applicable, but task-native outputs can be better matched to non-generative tasks.

Existing adaptation methods usually obtain task targets from human annotations or an external model and use them to post-train specialized models (Han et al.; Zheng et al., 2024). However, even for tasks with such output mismatches, an LLM can produce useful task signals itself. For example, although an LLM can-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

not directly generate dense image embeddings, it can describe an image in detail. Similarly, it can express grounding locations and class information through token sequences. This motivates an adaptation question: can these task-relevant signals, elicited from the model through an ill-suited output interface, be used to construct a more suitable task-native expression mechanism?

We address this question with **S**elf-**K**nowledge **R**e-expression (**SKR**). SKR first elicits pseudo-targets from a model's own generative behavior and then uses them to train task-native output layers together with supporting LoRA adapters. The adapters align the model's hidden states with the new output mechanism, enabling task-relevant information to be mapped efficiently to task-required outputs.

SKR uses no human-provided training annotations or outputs from external teacher models. It requires only unlabeled task inputs and a lightweight task configuration, which require much less effort and cost than acquiring external signals. An operator only needs to specify task prompts, output structures, training objectives, and any post-processing rules required to convert tokens into task-required formats. This makes SKR a task-general, cost-efficient adaptation recipe. Given a model and task data, the entire pipeline can run locally without sending data to external services. This reduces annotation costs and data exposure, making SKR particularly suitable for data-sensitive domains.

In summary, our contributions are as follows:

1. Knowledge expression as a new view of adaptation: We formulate the knowledge expression bottleneck as a mismatch between a model's generic NTP-based output mechanism and a task's native output requirements. This view highlights an overlooked source of LLM task performance and motivates a less costly approach to task adaptation.

2. The SKR framework: We introduce SKR that relies on a model's self-generated task signals to transform its expression mechanism into a task-native one. This method starts from unlabeled task inputs and requires neither human-provided training annotations nor external-teacher outputs.

3. Evaluation across output types and capability regimes: We evaluate SKR on dense retrieval, region localization, and imbalanced classification across several LLMs and MLLMs. Results show that SKR-adapted models consistently improve task effectiveness or inference efficiency, demonstrating the practical value of optimizing how LLMs express task-relevant information.

2 Problem Setup

2.1 Knowledge, Expression, and Task Interfaces

A model's ability to generate desired outputs stems from its parametric knowledge acquired during pre-training (Wang et al., 2024c; Wei et al., 2022). However, its performance on a task depends on both its task-relevant knowledge and how that knowledge is expressed. Let M_θ denote a pre-trained model with parameters θ. We use K_θ to denote the task-relevant knowledge and capabilities encoded in these parameters through pre-training.

We use E to denote the model's *expression mechanism*: the mapping from an input and the knowledge encoded in the model to an observable output. For example, the generic NTP-based expression mechanism, denoted by E_{ntp}, produces token sequences based on the input and the model's parametric knowledge. The relationship between a model and its expression mechanism can be written as

$$M_{\theta,E}(x) = E(x; K_\theta). \tag{1}$$

For a standard LLM, $E = E_{\text{ntp}}$ generates a token sequence $s = (w_1, \ldots, w_L)$. At each generation step, E_{ntp} produces a probability distribution over the vocabulary and generates the next token.

For a task T with required outputs in the space $\mathcal{Y}_T$, a task-native expression mechanism E_T instead produces the required output directly, such as a dense embedding, continuous values, or classification scores. In SKR, the original parameters θ remain frozen, while the LoRA parameters $\Delta\theta_T$ and additional task-specific output layers are trained. The adapted model is therefore

$$M_{\theta+\Delta\theta_T,E_T}(x) = E_T(x; K_{\theta+\Delta\theta_T}) \in \mathcal{Y}_T. \tag{2}$$

The transition from E_{ntp} to E_T constitutes the change in expression. The LoRA adapters $\Delta\theta_T$ align the model's hidden states with the new expression mechanism E_T. Knowledge re-expression refers to learning this change, relying on signals produced using the base model's own K_θ.

2.2 Limitations of Next-Token Prediction

Task performance is not determined by pre-trained capability alone; it also depends on how that capability is elicited and mapped to the required output (Gekhman et al., 2025). We study three representative mismatches between E_{ntp} and a task-native output space:

1. Information Retrieval (T_{IR}). Retrieval requires dense representations organized according to semantic proximity rather than decoded strings. NTP produces discrete tokens, while its hidden states are optimized for token generation rather than semantic retrieval. Consequently, these hidden states are poorly structured as retrieval representations.

2. Instruction-conditioned Region Localization (T_{Loc}). Expressing a bounding box as a coordinate string introduces sequential decoding, parsing, and possible syntax errors. Direct regression, which produces four continuous values in a single forward pass, is better suited to this task. In addition to accuracy, it can improve efficiency even when the base model already has strong localization ability.

3. Imbalanced Classification (T_{IC}). An LLM can generate label words such as *yes* or *no*. However, the resulting token probabilities are often poor proxies for task-oriented confidence scores. A classification head can provide scores optimized directly for the target decision rather than for vocabulary. We evaluate if task-specific scores improve discrimination under severe class imbalance.

These output types are foundational to applications including RAG (Wei et al., 2024; Li et al., 2025), object localization (Sui et al., 2024; Zhou et al., 2025), and anomaly detection (Xu et al., 2025; Deng et al., 2025). We use these three tasks as representative cases to demonstrate the applicability of the SKR recipe across different output types and tasks.

2.3 Definition of Self-Knowledge Re-expression

Let $\mathcal{D}_x = \{x_i\}_{i=1}^N$ be unlabeled inputs for task T. Adapting LLMs to specialized tasks typically requires extensive, high-quality annotations $\mathcal{D}_y = \{y_i\}_{i=1}^N$ obtained from human annotators (Chen et al., 2021) or external models (Wu et al., 2025). SKR removes this reliance by deriving its training targets from the model itself. It requires only a task configuration $\mathcal{C}_T$ specifying the task prompt, task-native expression mechanism E_T, training loss, and any parsing or post-processing rules π_T. Specifying this task-level information requires substantially less effort than collecting annotations and can be completed by someone familiar with the task within tens of minutes to a few hours.

Formally, Self-Knowledge Re-expression is defined as

$$M_{\theta, E_{\text{ntp}}} \xrightarrow[\mathcal{D}_x, \mathcal{C}_T]{\text{SKR}} M_{\theta+\Delta\theta_T, E_T}. \tag{3}$$

During this transition, the base model expresses its own knowledge K_θ through E_{ntp} to generate task-specific training signals from $\mathcal{D}_x$ under the task configuration $\mathcal{C}_T$. These signals are then used to train the LoRA parameters $\Delta\theta_T$ and the task-specific output layers to acquire the new expression E_T.

In Equation 3, *self-knowledge* describes the source of training signals: they are generated by the model being adapted rather than provided externally. *Re-expression* describes the resulting transition from the NTP expression mechanism E_{ntp} to the task-native expression mechanism E_T. The two stages underlying Equation 3, self-annotation and knowledge re-expression, are detailed in §3.

SKR differs from standard knowledge distillation and teacher-student adaptation in two respects: (1) SKR uses the adapted model's own knowledge K_θ to generate task signals, without receiving external supervision; and (2) it uses these signals to change the expression mechanism ($E_{\text{ntp}} \rightarrow E_T$), rather than to keep E_{ntp}. Although SKR shares the use of pseudo-targets with self-training, its objective is to construct a task-native expression mechanism rather than to improve token generation.

3 Method

Equation 3 defines the overall SKR transition. This section details its implementation in two steps, as illustrated in Figure 2. The task configuration $\mathcal{C}_T$ typically contains four components: (1) a task prompt P_T that queries the LLM for the given task; (2) a task-native expression mechanism E_T, including the structure of the task-specific output layers; (3) a task loss $\mathcal{L}_T$; and (4) a post-processing

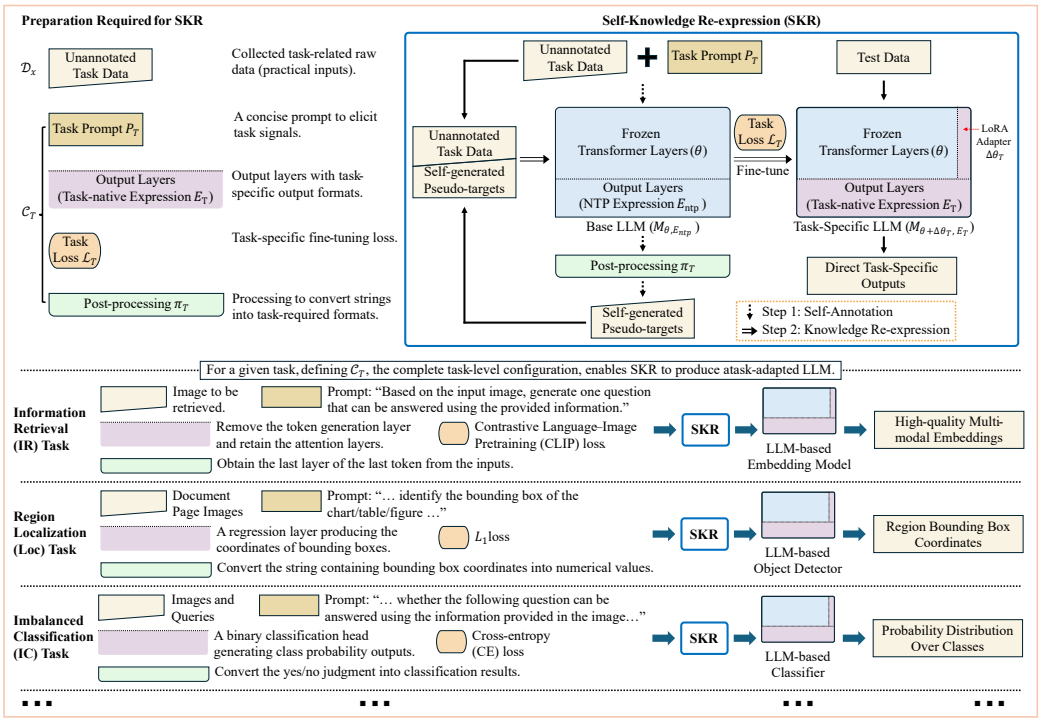


Figure 2: SKR implemented across three representative tasks. It requires only unlabeled task inputs $\mathcal{D}_x$ and a task configuration $\mathcal{C}_T$ to implement task adaptation.

rule π_T that converts generated tokens into the task-required format. Most of these task-level design choices are also required when fine-tuning a task-specific model with annotated data. SKR removes the most costly component, data annotations, by using the model itself as the source of training signals and applying knowledge re-expression to convert these signals into task-native capabilities.

1. Self-Annotation: Eliciting task signals through E_{ntp}. For each unlabeled input $x_i \in \mathcal{D}_x$, the task prompt P_T elicits a target-related token sequence from the base model. In this process, the model expresses its own task-relevant knowledge K_θ through E_{ntp}. The parsing or post-processing rule π_T then converts the generated sequence into a pseudo-target $\hat{y}_i$. The resulting pseudo-target set is

$$\hat{\mathcal{D}}_y = \{\pi_T(E_{\text{ntp}}(x_i, P_T; K_\theta))\}_{i=1}^{N}. \tag{4}$$

In $\hat{\mathcal{D}}_y$, an individual pseudo-target may be incorrect. But at the population level, $\hat{\mathcal{D}}_y$ reflects the task-relevant knowledge already encoded in K_θ and can therefore provide a useful overall training direction despite errors in individual examples.

2. Knowledge Re-expression: Adapting to E_T. The pairs $(\mathcal{D}_x, \hat{\mathcal{D}}_y)$ are used to optimize the task-native loss $\mathcal{L}_T$. The pre-trained parameters θ remain frozen, while the LoRA parameters $\Delta\theta_T$ and the task-specific output layers implementing E_T are trained jointly. This step is formalized as

$$M_{\theta,E_{\text{ntp}}} \xrightarrow[(\mathcal{D}_x,\hat{\mathcal{D}}_y),\,\mathcal{L}_T]{\text{train } \Delta\theta_T \text{ and } E_T} M_{\theta+\Delta\theta_T,E_T}. \tag{5}$$

Figure 2 provides concrete implementations for three representative tasks. The four components presented in the figure constitute the complete set of task-level information supplied to SKR. Their compactness also shows that specifying $\mathcal{C}_T$ is substantially lighter than constructing annotations. As LLMs and multimodal LLMs (MLLMs) use the same autoregressive token interface, we focus primarily on MLLMs for more complex tasks while retaining a text-only retrieval variant.

3.1 SKR for Information Retrieval (T_{IR})

The first case covers text-to-text (paragraph) and text-to-image (chart/figure) retrieval, where the objective is to identify an item containing the answer to a query. Although hidden states from

an NTP-trained model can be pooled into vectors, they are not explicitly optimized for semantic retrieval (Jiang et al., 2024). Existing systems commonly fine-tune representations using query–item pairs created by humans or stronger models (Zhou et al., 2024; Koukounas et al., 2024).

SKR instead prompts the same model to generate a query from each raw paragraph or image, forming synthetic query–item pairs for contrastive learning (Radford et al., 2021). During re-expression, we bypass the vocabulary projection and use the last-layer hidden state of the final input token as the dense representation. Fine-tuning with a contrastive loss transforms the model from $M_{\theta,E_{\text{ntp}}}$ to $M_{\theta+\Delta\theta_{\text{IR}},E_{\text{IR}}}$, enabling SKR-adapted models to surpass advanced IR baselines.

3.2 SKR FOR REGION LOCALIZATION (T_{Loc})

The study of T_{Loc} demonstrates how SKR streamlines multi-step localization into a one-pass operation. Practical document localization often requires flexible granularity. For example, extracting a chart may require including its title and source to retain context. Traditional detectors, like YOLO (Redmon et al., 2016), lack zero-shot instruction flexibility. Although MLLMs can adjust detection via prompting, they often struggle with composite targets (e.g., figure + title). Current workarounds split detection into multiple sub-tasks and merge outputs, which significantly increases latency and risk of failures.

Through SKR, we run such multi-step localization on a small unlabeled subset to generate coordinate pseudo-targets. A regression head implementing E_{Loc} then outputs four normalized bounding-box values and is optimized jointly with LoRA using L_1 loss. The adapted model $M_{\theta+\Delta\theta_{\text{Loc}},E_{\text{Loc}}}$ replaces multi-turn decoding with a single forward pass. For models without strong coordinate generation, fuzzy-localization prompting or post-processing with a local OCR tool can convert their semantic responses into approximate coordinate supervision. SKR thereby transforms semantic image understanding expressed through NTP into explicit grounding capability.

3.3 SKR FOR IMBALANCED CLASSIFICATION (T_{IC})

The third task is an image–text matching problem: determining whether an image contains information relevant to a query. Since valid matches are rare, the data is highly imbalanced. The base model supplies either generated labels or scores derived from label tokens, whereas a classification head implementing E_{IC} produces a score optimized directly for the target classes.

In self-annotation, we prompt the model to output an explicit *yes* or *no* for each query–image pair and map that response to a binary pseudo-label. During re-expression, a classification head produces classification logits and is optimized with cross-entropy. We then calculate AUROC and AUPRC, which are informative under class imbalance (Hu et al., 2020; Wang et al., 2022). These metrics can also be computed from token scores, but task-specific scores provide remarkably better discrimination.

4 EXPERIMENTAL SETTINGS

4.1 DATASET AND EVALUATION METRICS

We evaluate SKR using the following datasets. For all datasets, SKR uses only unannotated raw data. Annotations are strictly reserved for evaluation. More dataset details are provided in Appendix A.

Financial Document Dataset: This industrial dataset contains over 100,000 reports with millions of tables and charts. It provides a rigorous environment for T_{IR}, T_{Loc}, and T_{IC}, reflecting real-world financial analysis complexities. For each task, SKR uses only a small randomly sampled unlabeled subset ($\approx$1,000 examples), while evaluation uses manually verified test sets. The three tasks are retrieving paragraphs or images for textual queries (T_{IR}), localizing charts and tables in document pages (T_{Loc}), and classifying query–image pairs as valid or invalid (T_{IC}).

MMDocRAG, DocVQA-2020, and SciMMIR (Dong et al., 2025; Mathew et al., 2021; Wu et al., 2024): We implement SKR using only unlabeled MMDocRAG inputs and evaluate it on all three multimodal retrieval benchmarks, enabling comparison with published retrievers and testing generalization to the unseen DocVQA-2020 and SciMMIR datasets.

CUB_200_2011 Dataset (Wah et al., 2011): For T_{Loc}, and T_{IC}, we further evaluate SKR on this open-source dataset. It contains 200 bird species with fine-grained bounding-box annotations.

We evaluate each task using a comprehensive set of metrics, with definitions detailed in Appendix B. We report mean reciprocal rank (MRR) and Recall@k (R@k) for T_{IR}; intersection over union (IoU) and inference time (Time) for T_{Loc}; and accuracy (Acc), area under the ROC curve (AUROC), area under the precision-recall curve (AUPRC), and Time for T_{IC}.

4.2 Models and Baselines

We run four LLMs for text tasks and three MLLMs for multi-modal tasks. LLMs include Mistral-7B-Instruct-v0.2 (Mistral0.2; Jiang et al. 2023), Phi-3-mini-128k-instruct (Phi3; Abdin et al. 2024), Qwen2-7B-Instruct (Qwen2; Team et al. 2024), and Llama-3.1-8B (Llama3.1; Grattafiori et al. 2024). MLLMs include LLaVA-v1.6-Mistral-7B (LLaVA1.6; Liu et al. 2023), Phi-3-vision-128k-instruct (Phi3-VL; Abdin et al. 2024), and Qwen2-VL-7B-Instruct (Qwen2-VL; Wang et al. 2024b).

Our primary controlled baseline is the same base model before SKR. We include E5-V (Jiang et al., 2024), a training-free representation method, and several advanced LLM-based embedding models (Nomic Team 2025, OpenAI, and leading models reported by MMDocRAG) for retrieval context. They are strong baselines trained on large annotated corpora, a substantially more resource-intensive setting than SKR. We additionally run self-SFT, which retains a token-oriented objective, and head-only, which freezes the backbone and trains only the expression layers, for ablation baselines.

4.3 SKR Implementation Details

The *Self-Annotation* phase is tailored to the base model's observable capabilities. For T_{IR} and T_{IC}, we directly prompt the model to generate input-related queries or assess image-text relevance. For T_{Loc}, the strategy differs by model capability. We directly prompt Qwen2-VL to output coordinates because it possesses native grounding capability. Models with semantic image understanding but no native coordinate generation (LLaVA1.6 and Phi3-VL) instead use "fuzzy localization", which recursively crops inputs and queries object presence, to construct approximate bounding boxes. We also evaluate lightweight local OCR as an explicitly task-specific post-processing tool. Further details on self-annotation, prompts, task head structures, and hyper-parameters are provided in Appendix C.

5 Results and Discussion

We compare SKR-adapted models with their base versions and advanced baselines across T_{IR}, T_{Loc}, and T_{IC} in §5.1. §5.2 then reports analysis of ablation studies. Appendix D provides additional public-dataset results, Appendix E analyzes model scale and cross-model supervision, Appendix F reports full head-only ablations, and Appendix G describes representation shifts using CKA.

Table 1: T_{IR} performance comparison. SKR results are shown in bold. The published embedding systems are contextual baselines and do not share the same training regime.

Text-to-text Retrieval

Model		MRR	R@1	R@3	R@10
OpenAI ada-002		0.420	0.322	0.460	0.598
Nomic-text-v2		0.549	0.454	0.600	0.728
Mistral-0.2	Base	0.108	0.052	0.116	0.204
	E5-V	0.223	0.144	0.250	0.386
	SFT	0.076	0.036	0.078	0.152
	SKR	**0.622**	**0.510**	**0.702**	**0.822**
Phi-3	Base	0.098	0.068	0.096	0.146
	E5-V	0.209	0.142	0.230	0.338
	SFT	0.082	0.054	0.084	0.124
	SKR	**0.632**	**0.518**	**0.718**	**0.826**
Qwen2	Base	0.115	0.062	0.130	0.230
	E5-V	0.270	0.196	0.294	0.416
	SFT	0.063	0.032	0.056	0.136
	SKR	**0.674**	**0.566**	**0.756**	**0.860**

Text-to-image Retrieval

Model		MRR	R@1	R@3	R@10
Nomic-mm-7b		0.669	0.560	0.745	0.860
LLaVA-1.6	Base	0.014	0.000	0.010	0.035
	E5-V	0.197	0.150	0.205	0.305
	SFT	0.031	0.020	0.025	0.065
	SKR	**0.552**	**0.445**	**0.625**	**0.730**
Phi-3-VL	Base	0.071	0.050	0.065	0.125
	E5-V	0.203	0.150	0.210	0.300
	SFT	0.027	0.020	0.030	0.035
	SKR	**0.556**	**0.450**	**0.630**	**0.750**
Qwen2-VL	Base	0.044	0.030	0.035	0.090
	E5-V	0.291	0.210	0.340	0.450
	SFT	0.055	0.035	0.055	0.085
	SKR	**0.691**	**0.595**	**0.755**	**0.875**

5.1 TASK IMPROVEMENT OF SKR

Results on T_{IR}. Table 1 shows that SKR substantially improves T_{IR} of both LLMs and MLLMs. Representations taken directly from the base models perform poorly. While E5-V partially mitigates this limitation, its results remain insufficient. SKR, using only self-generated signals, adapts the models to industrial level encoding models. The strongest results, obtained by SKR-adapted Qwen2 models, even surpass advanced models trained with much larger supervised or synthetic corpora.

Additional Llama3.1 results (Appendix D.1) and evaluations on MMDocRAG, DocVQA, and SciMMIR (Appendix D.2) show the same trend and provide evidence of cross-dataset transfer. SKR consistently outperforms a range of SOTA retrieval models. These results demonstrate that LLMs possess substantial latent task-relevant capabilities and that SKR can unlock this potential efficiently.

Results on T_{Loc}. Table 2 shows that SKR can elicit latent task capability and improve inference efficiency. LLaVA1.6 and Phi3-VL perform poorly because their native localization ability is weak despite their semantic image understanding. SKR elicits this latent capability through fuzzy localization and re-expresses it as direct bounding-box regression, consistently improving both models. Stronger post-processing through the OCR-assisted strategy, which uses lightweight local OCR without extra LLMs, further leads to substantially better performance.

Table 2: T_{Loc} performance comparison. SKR results are in bold; best results within each model are underlined. Runtime is reported in min/100 cases.

Model		IoU	Time
LLaVA1.6	Base (Fuzzy)	0.142	41.0
	SFT (Fuzzy)	0.065	6.4
	SKR (Fuzzy)	**0.266**	**1.7**
	Base (OCR)	0.599	10.3
	SKR (OCR)	**0.614**	**1.6**
Phi3-VL	Base (Fuzzy)	0.164	35.0
	SFT (Fuzzy)	0.056	7.1
	SKR (Fuzzy)	**0.232**	**0.8**
	Base (OCR)	0.336	30.7
	SKR (OCR)	**0.344**	**0.9**
Qwen2-VL	Base (One-turn)	0.628	6.8
	Base (Multi-turn)	0.739	20.4
	SFT (Multi-turn)	0.741	6.5
	SKR (Multi-turn)	**0.726**	**4.8**

Qwen2-VL represents a different capability regime. It already possesses strong grounding ability. The benefit of SKR is to express the existing capability more efficiently. SKR compresses multi-turn generation into a single regression pass, substantially reducing inference time. The results on CUB_200_2011 in Appendix D.3 show the same pattern: SKR elicits localization ability from weaker models while improving the efficiency of strong models.

Results on T_{IC}. In this setting, accuracy can be deceptive because negative cases dominate. Table 3 shows that the task-specific SKR score generally provides stronger ranking discrimination than the token-based alternatives. To ensure a rigorous baseline, we tried two probability extraction methods for base models: assigning 100% to the generated class and getting softmax probabilities for specific tokens. We report the superior of these two as the base model results.

Table 3: T_{IC} performance comparison. SKR results are in bold. Runtime is reported in min/100 cases.

Model		Acc	AUROC	AUPRC	Time
LLaVA-1.6	Base	0.930	0.894	0.531	3.8
	SFT	0.835	0.892	0.381	1.4
	SKR	**0.975**	**0.991**	**0.879**	**1.3**
Phi-3-VL	Base	0.770	0.850	0.287	4.7
	SFT	0.945	0.912	0.850	7.1
	SKR	**0.935**	**0.981**	**0.933**	**0.7**
Qwen2-VL	Base	0.935	0.875	0.532	1.1
	SFT	0.960	0.939	0.746	0.9
	SKR	**0.960**	**0.984**	**0.868**	**0.7**

While accuracy gains are modest, AUROC and AUPRC improve substantially. All three models gain at least 0.1 AUROC and 0.3 AUPRC, with Phi-3-VL increasing from 0.287 to 0.933 AUPRC, indicating better discrimination. SKR also reduces runtime by over 30%. The trend is further verified on CUB_200_2011 (Appendix D.3).

5.2 ABLATION STUDIES

We use three sets of controlled comparisons to examine the SKR pipeline: direct quality measurement of self-generated targets, replacement of self-generated targets with an external teacher, and component ablations over the output interface and LoRA adapters. These experiments assess whether the model's own knowledge provides sufficient task signals and quantify the importance of SKR designs.

5.2.1 QUALITY OF SELF-GENERATED TARGETS

We evaluate the self-generated targets against the CUB_200_2011 test-set ground truth, which is used only for this analysis and never for SKR training. For T_{Loc}, the pseudo-targets generated by Qwen2-VL, LLaVA-1.6, and Phi-3-VL achieve IoUs of 0.684, 0.271, and 0.211, respectively; for T_{IC}, their AUPRCs are 0.363, 0.174, and 0.239. Although imperfect, these targets show nontrivial agreement with the ground truth and provide useful population-level training signals. We further examine the effect of target quality through the external-teacher comparison below and the model-scale analysis in Appendix E.

5.2.2 SOURCE OF TASK SIGNALS: SELF ANNOTATION VS. EXTERNAL SUPERVISION

We replace the self-generated targets with GPT-4o-generated targets for T_{IR} and T_{Loc}, while keeping the adapted model and re-expression procedure fixed. GPT-4o is generally stronger than the evaluated models on multimodal benchmarks (Zhu et al., 2025). Figure 3 compares the two signal sources.

For T_{IR}, self-generated targets match or outperform GPT-4o targets. An external teacher therefore provides no additional benefit, whereas adopting a retrieval-native expression mechanism produces large gains. This supports the expression mechanism as the main bottleneck in this setting when the model's own task signals are sufficient.

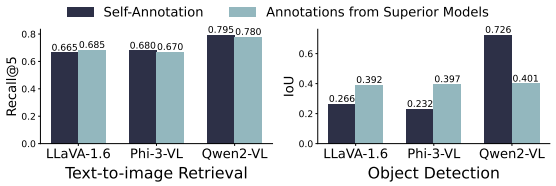


Figure 3: Performance comparison of LLMs fine-tuned using self-annotations versus external annotations across T_{IR} and T_{Loc}.

For T_{Loc}, the outcome depends on the annotator's grounding capability. GPT-4o-generated targets benefit LLaVA-1.6 and Phi-3-VL, whose self-generated coordinates are weak, while Qwen2-VL benefits more from its own targets. Despite GPT-4o's overall strength, its grounding performance on our test set (0.414 IoU) is lower than that of Qwen2-VL. Therefore, broad model strength does not guarantee higher-quality task-specific annotations. The cross-model experiments within the Gemma-3 family in Appendix E further distinguish annotation quality from the capacity of the adapted model.

Overall, an external teacher is unnecessary when a model's own outputs provide relatively sufficient task signals but can help when self-annotation reveals a capability gap. These findings further suggest that SKR can unlock specialized capabilities even in strong LLMs.

5.2.3 COMPONENT ANALYSIS: EXPRESSION MECHANISM AND LORA ADAPTERS

We first examine whether the gains of SKR can be obtained by standard SFT using the same self-generated targets. SFT retains E_{ntp} rather than adopting E_T. Results are summarized in the SFT rows of Tables 1, 2, and 3. On T_{IR}, SFT performs poorly, yielding only marginal gains in text-to-image retrieval and even degrading text-to-text retrieval. On T_{Loc}, SFT fails to elicit useful localization from LLaVA1.6 and Phi3-VL. For Qwen2-VL, which already has strong grounding capability, SFT maintains the base model's ability but still suffers from inefficient token generation at inference. On T_{IC}, SKR consistently achieves higher AUPRC than SFT for all three models. These comparisons demonstrate that self-generated supervision alone is insufficient. Its task signals become substantially more useful when the model is adapted to a task-native expression mechanism.

We next compare head-only with full SKR to assess the contribution of LoRA. Head-only trains only the task-native output layers, whereas full SKR additionally trains $\Delta\theta_T$. Table 4 reports the Qwen2-VL results, with complete results in Appendix F.

Adding LoRA improves all three tasks. The task-specific output layers alone cannot fully use the unadapted hidden states. Given the scale and depth of LLMs, a change in expression cannot be achieved by modifying only a small number of parameters at the output. The hidden states across the model layers must also adapt to support the new expression mechanism. Appendix G further analyzes the hidden-state changes introduced by LoRA.

Table 4: LoRA adapter ablation for Qwen2-VL.

Method	T_{IR} R@10	T_{Loc} IoU	T_{IC} AUPRC
Head-only	0.652	0.491	0.492
Full SKR	**0.877**	**0.692**	**0.545**

6 Related Work

Knowledge acquisition and task adaptation. The evolution of LLMs has been largely defined by the scaling hypothesis, which posits that model performance improves predictably with increases in parameter count and training data (Hestness et al., 2017; Kaplan et al., 2020). Subsequent research has emphasized that optimizing the volume and quality of training tokens to enhance knowledge acquisition is as critical as scaling model size (Brown et al., 2020; Hoffmann et al., 2022). Retrieval-augmented generation (RAG) further expands model capabilities by externalizing knowledge into non-parametric memory, improving performance on knowledge-intensive tasks (Lewis et al., 2020). However, despite these advances in knowledge acquisition, LLMs still exhibit significant performance gaps on non-generative tasks (Grabuloski et al., 2025; Song et al., 2025). The prevailing strategy for bridging this gap remains fine-tuning models using task-specific datasets, demonstrations, preferences, or reward signals (Kaufmann et al., 2023; Koukounas et al., 2024; Xie et al., 2023).

Self-generated supervision and synthetic data. As the cost of training data and the data exposure risks increase, research has shifted toward using task-relevant signals available in pre-trained models (Chen et al., 2024). Current post-training paradigms have explored self-reflection (Qiu & Yang, 2024; Yuan et al., 2024), weak supervision (Yu et al., 2021), pseudo-labeling, and self-generated rewards (Shao et al., 2025) to improve task performance. SKR belongs to this broad family but differs from them on the training objectives. Existing methods refine LLMs' generation while retaining their NTP interface. SKR instead adapts the model to a different, task-native output mechanism.

Task-native outputs for generative models. A growing body of evidence suggests that the NTP paradigm is not a panacea for all tasks (Bachmann & Nagarajan, 2024). For non-generative tasks such as information retrieval and classification, task-oriented output mechanisms can outperform autoregressive token generation (Wang & Ma, 2024; Ma et al., 2024; Wang et al., 2024a). Encoder and representation-learning methods have long used classification, regression, and embedding heads specialized for downstream tasks (Karpukhin et al., 2020; Zhang et al., 2022). Recent work similarly converts LLMs into embedding models through hidden-state readouts or contrastive adaptation (Jiang et al., 2024; Ma et al., 2024; Faysse et al., 2025), while regression-aware objectives can improve numerical prediction by autoregressive models (Lukasik et al., 2025). These findings motivate adapting not only what an LLM generates, but also how it expresses task-relevant information.

Building on these directions, SKR combines same-model self-generated supervision with task-native adaptation. The model's generative outputs provide pseudo-targets for learning a different output mechanism. This integration improves task performance and efficiency without external annotations.

7 Conclusion, Limitation, and Future Work

LLM performance can be limited not only by what a model knows, but also by how its task-relevant knowledge is expressed. We introduced SKR as a task-general framework for studying and addressing this second adaptation axis. SKR elicits task-relevant signals from a model's own generative behavior and uses them to adapt the model to a task-native output interface. Across three tasks, SKR effectively improves performance and efficiency, even surpassing specialized models.

This work has two primary limitations. First, resource constraints prevented evaluation on the latest open-source models. Nevertheless, our results already surpass many state-of-the-art models. Our goal is to provide evidence for the practical importance of the expression bottleneck rather than to pursue the performance upper bound. The results show that diverse LLMs possess substantial unexploited capabilities, allowing even modest models to outperform highly specialized counterparts through SKR. Second, SKR requires a task-specific configuration $\mathcal{C}_T$, which involves some design effort. However, its components follow standard task-adaptation practices and are lightweight to specify. Defining $\mathcal{C}_T$ requires substantially less effort than constructing instance-level annotations. The entire SKR pipeline can be executed locally without exposing task data to external services.

Looking forward, a promising direction is the full automation of SKR, in which LLMs autonomously construct $\mathcal{C}_T$ to further reduce human intervention. More broadly, this work establishes knowledge re-expression as a new adaptation perspective. Our findings suggest a path toward improving advanced AI systems without continuously scaling training data or model parameters, offering a practical, cost-efficient, and data-local approach to deploying models in sensitive real-world applications.

AI USE STATEMENT

Generative AI tools were used in two roles in this work. First, as an integral part of the proposed SKR framework, the evaluated LLMs and MLLMs were used to generate task-specific pseudo-targets from unlabeled data. GPT-4o was additionally used as an external teacher in the corresponding controlled comparison. These AI-generated targets, their construction procedures, and their roles in training are explicitly described in the paper.

Second, generative AI tools assisted with manuscript preparation. They were primarily used to correct grammar and typographical errors, improve readability and coherence, and suggest revisions to the organization and presentation of conceptual definitions and experimental discussions. Some descriptive factual passages, particularly dataset descriptions, were initially drafted with AI assistance. The authors manually verified these passages against the cited sources, dataset records, experimental settings, and reported results.

All AI-assisted content was reviewed and edited by the authors. The authors made the final decisions regarding the methodology, experiments, analysis, claims, and presentation, and take full responsibility for all content in this paper.

REPRODUCIBILITY STATEMENT

We are unable to release the implementation code. To support independent reproduction, we provide detailed descriptions of the SKR pipeline and its task-specific implementations in §3. The datasets, evaluated models, baselines, metrics, and experimental settings are reported in §4. Appendix C further provides the complete prompts, self-annotation procedures, post-processing rules, task-specific output-layer structures, training objectives, and hyperparameters required for reimplementation.

In addition to the industrial financial-document dataset, we report extensive experiments on public benchmarks, including MMDocRAG, DocVQA-2020, SciMMIR, and CUB_200_2011. Additional public-dataset results are provided in Appendix D. Each public dataset is accompanied by a citation to its original paper or source, enabling independent access and evaluation under the corresponding benchmark protocols.

REFERENCES

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

Gregor Bachmann and Vaishnavh Nagarajan. The pitfalls of next-token prediction. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 2296–2318, 2024.

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

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan R Routledge, et al. Finqa: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3697–3711, 2021.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. In *International Conference on Machine Learning*, pp. 6621–6642. PMLR, 2024.

Huilin Deng, Hongchen Luo, Wei Zhai, Yanming Guo, Yang Cao, and Yu Kang. Vmad: Visual-enhanced multimodal large language model for zero-shot anomaly detection. *IEEE Transactions on Automation Science and Engineering*, 2025.

Kuicai Dong, CHANG YUJING, Shijie Huang, Yasheng Wang, Ruiming Tang, and Yong Liu. Benchmarking retrieval-augmented multimomal generation for document question answering. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi,

and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc., 2025. doi: 10.52202/085713-0619. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/1a93178950e92fd2e7b7448f7d68fd7d-Paper-Datasets_and_Benchmarks_Track.pdf.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. Colpali: Efficient document retrieval with vision language models. In *ICLR*, 2025.

Zorik Gekhman, Eyal Ben David, Hadas Orgad, Eran Ofek, Yonatan Belinkov, Idan Szpektor, Jonathan Herzig, and Roi Reichart. Inside-out: Hidden factual knowledge in llms. *arXiv preprint arXiv:2503.15299*, 2025.

Marko Grabuloski, Aleksandar Karadimce, and Anis Sefidanoski. Enhancing language models with retrieval-augmented generation a comparative study on performance. *WSEAS Transactions on Information Science and Applications*, 22:272–297, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *Transactions on Machine Learning Research*.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pp. 30016–30030, 2022.

Wenpeng Hu, Mengyu Wang, Qi Qin, Jinwen Ma, and Bing Liu. Hrn: A holistic approach to one class learning. *Advances in neural information processing systems*, 33:19111–19124, 2020.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Ting Jiang, Minghui Song, Zihan Zhang, Haizhen Huang, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, and Fuzhen Zhuang. E5-v: Universal embeddings with multimodal large language models. *arXiv preprint arXiv:2407.12580*, 2024.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, 2020.

Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback. *arXiv preprint arXiv:2312.14925*, 2023.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMlR, 2019.

Andreas Koukounas, Georgios Mastrapas, Michael Günther, Bo Wang, Scott Martens, Isabelle Mohr, Saba Sturua, Mohammad Kalim Akram, Joan Fontanals Martínez, Saahil Ognawala, et al. Jina clip: Your clip model is also your text retriever. *arXiv preprint arXiv:2405.20204*, 2024.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Ying Li, Mengyu Wang, Miguel de Carvalho, Sotirios Sabanis, and Tiejun Ma. FinGEAR: Financial mapping-guided enhanced answer retrieval. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 7239–7255, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.382. URL https://aclanthology.org/2025.findings-emnlp.382/.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Michal Lukasik, Zhao Meng, Harikrishna Narasimhan, Yin-Wen Chang, Aditya Krishna Menon, Felix Yu, and Sanjiv Kumar. Better autoregressive regression with llms via regression-aware fine-tuning. In *ICLR*, 2025.

Xueguang Ma, Sheng-Chieh Lin, Minghan Li, Wenhu Chen, and Jimmy Lin. Unifying multimodal retrieval via document screenshot embedding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 6492–6505, 2024.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2200–2209, 2021.

Junyan Qiu and Yiping Yang. Training large language models to follow system prompt with self-supervised fine-tuning. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2024.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

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

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, et al. Spurious rewards: Rethinking training signals in rlvr. *arXiv preprint arXiv:2506.10947*, 2025.

Zirui Song, Bin Yan, Yuhan Liu, Miao Fang, Mingzhe Li, Rui Yan, and Xiuying Chen. Injecting domain-specific knowledge into large language models: a comprehensive survey. *arXiv preprint arXiv:2502.10708*, 2025.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pp. 645–654, 2024.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Nomic Team. Nomic embed multimodal: Interleaved text, image, and screenshots for visual document retrieval, 2025. URL https://nomic.ai/blog/posts/nomic-embed-multimodal.

Qwen Team et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2(3), 2024.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Mengyu Wang and Tiejun Ma. Mana-net: Mitigating aggregated sentiment homogenization with news weighting for enhanced market prediction. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 2379–2389, 2024.

Mengyu Wang, Yijia Shao, Haowei Lin, Wenpeng Hu, and Bing Liu. Cmg: A class-mixed generation approach to out-of-distribution detection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 502–518. Springer, 2022.

Mengyu Wang, Shay B Cohen, and Tiejun Ma. Modeling news interactions and influence for financial market prediction. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 3302–3314, 2024a.

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

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. Knowledge editing for large language models: A survey. *ACM Computing Surveys*, 57(3):1–37, 2024c.

Cong Wei, Yang Chen, Haonan Chen, Hexiang Hu, Ge Zhang, Jie Fu, Alan Ritter, and Wenhu Chen. Uniir: Training and benchmarking universal multimodal information retrievers. In *European Conference on Computer Vision*, pp. 387–404. Springer, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Siwei Wu, Yizhi Li, Kang Zhu, Ge Zhang, Yiming Liang, Kaijing Ma, Chenghao Xiao, Haoran Zhang, Bohao Yang, Wenhu Chen, et al. Scimmir: Benchmarking scientific multi-modal information retrieval. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12560–12574, 2024.

Yifan Wu, Lutao Yan, Yizhang Zhu, Yinan Mei, Jiannan Wang, Nan Tang, and Yuyu Luo. Boosting text-to-chart retrieval through training with synthesized semantic insights. *arXiv preprint arXiv:2505.10043*, 2025.

Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. Pixiu: a large language model, instruction data and evaluation benchmark for finance. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 33469–33484, 2023.

Jiacong Xu, Shao-Yuan Lo, Bardia Safaei, Vishal M Patel, and Isht Dwivedi. Towards zero-shot anomaly detection and reasoning with multimodal large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 20370–20382, 2025.

Yue Yu, Simiao Zuo, Haoming Jiang, Wendi Ren, Tuo Zhao, and Chao Zhang. Fine-tuning pre-trained language model with weak supervision: A contrastive-regularized self-training approach. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1063–1077, 2021.

Huizhuo Yuan, Zixiang Chen, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning of diffusion models for text-to-image generation. *Advances in Neural Information Processing Systems*, 37: 73366–73398, 2024.

Shunyu Zhang, Yaobo Liang, Ming Gong, Daxin Jiang, and Nan Duan. Multi-view document representation learning for open-domain dense retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5990–6000, 2022.

Zheng Zhao, Yftah Ziser, and Shay B Cohen. Layer by layer: Uncovering where multi-task learning happens in instruction-tuned large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 15195–15214, 2024.

Jiawei Zheng, Hanghai Hong, Xiaoli Wang, Jingsong Su, Yonggui Liang, and Shikai Wu. Fine-tuning large language models for domain-specific machine translation. *CoRR*, 2024.

Junjie Zhou, Zheng Liu, Shitao Xiao, Bo Zhao, and Yongping Xiong. Vista: Visualized text embedding for universal multi-modal retrieval. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3185–3200, 2024.

Yitong Zhou, Mingyue Cheng, Qingyang Mao, Jiahao Wang, Feiyang Xu, and Xin Li. Enhancing table recognition with vision llms: a benchmark and neighbor-guided toolchain reasoner. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 2503–2511, 2025.

Zifeng Zhu, Mengzhao Jia, Zhihan Zhang, Lang Li, and Meng Jiang. Multichartqa: Benchmarking vision-language models on multi-chart problems. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 11341–11359, 2025.

Appendix Overview This Appendix provides supplementary material for the methodology, experimental setup, and extended analysis. We first present dataset specifications, evaluation-metric definitions, and implementation details in Appendices A, B, and C, respectively. Subsequently, we present extensive additional experimental results to offer deeper insights into the SKR method. These include:

- Performance evaluations across the three core tasks using the open-source MMDocRAG, DocVQA-2020, SciMMIR, and CUB_200_2011 datasets (Appendix D).
- An investigation into the impact of model scaling on SKR performance (Appendix E).
- Additional ablation studies regarding the necessity of LoRA adapters (Appendix F).
- A comparative study of latent representation shifts using CKA analysis (Appendix G).

A Dataset Details

We evaluate SKR using a large-scale industrial document dataset and four public benchmarks: MMDocRAG, DocVQA-2020, SciMMIR, and CUB_200_2011. Together, these datasets evaluate SKR in both practical industrial applications and standardized academic settings.

A.1 Financial Document Dataset

All three task types (T_{IR}, T_{Loc}, T_{IC}) are implemented using our proprietary financial document dataset. As highlighted in §4.1, we use only a small, randomly sampled subset of raw, unannotated data to activate the SKR process. To ensure rigorous evaluation, all ground-truth labels in the test sets were established through the consensus of at least two human experts.

Information Retrieval (T_{IR}): For text-to-text retrieval, we sample 1,000 paragraphs as unannotated training data for self-annotation. For testing, we construct a candidate pool of 5,000 distinct paragraphs, within which we identify 500 ground-truth query-paragraph pairs verified by humans. For

text-to-image retrieval, we sample 1,100 images (including charts, tables, and figures) as the unannotated task data. The test set consists of a candidate pool of 5,000 images, with 200 human-verified query-image pairs.

Efficient Region Localization (T_{Loc}): We sample 800 document pages, each containing a chart, table, or figure, as the unannotated task data $\mathcal{D}_x$. For the test set, we separately sample 200 pages and manually annotate the bounding boxes of the target objects.

Imbalanced Classification (T_{IC}): We reuse the 1,100-image set from T_{IR} as the unannotated training data. The test set is derived from the T_{IR} test set and casts rare valid matches as anomalies. It comprises 200 samples: 20 positive cases (images matching the query) and 180 negative cases (images irrelevant to the query), all manually verified.

A.2 MMDocRAG

MMDocRAG is one of the latest multi-modal benchmarks designed to evaluate the retrieval and reasoning capabilities of MLLMs (Dong et al., 2025). It encompasses 14,826 images across 222 documents and includes 4,055 expert-annotated question-answer pairs. The original benchmark evaluates query-image retrieval using several specialized retrievers, including DSE models (Ma et al., 2024) and ColPali models (Faysse et al., 2025). These published systems provide strong retrieval context. These compared models are LLMs specifically fine-tuned on massive multi-modal datasets.

To validate SKR's competitiveness, we adhered strictly to the official MMDocRAG test protocols. However, our training regime is significantly more rigorous: SKR starts from base LLMs and uses only the raw images from the development set to fine-tune models. We deliberately ignore all expert-annotated questions to maintain a strictly annotation-free adaptation process. As shown in Appendix D.2, SKR-adapted models outperform all advanced models evaluated in the original benchmark.

A.3 DocVQA-2020 and SciMMIR

DocVQA-2020 (Mathew et al., 2021) includes documents spanning multiple decades and diverse industries. It consists of 50,000 questions defined on 12,000+ document images. SciMMIR (Wu et al., 2024) is a specialized scientific domain dataset that presents a significant retrieval challenge, requiring the model to identify one relevant figure from a large candidate pool of 16,263 items.

To evaluate cross-dataset generalization, we use only the test sets of these two benchmarks. The evaluation is conducted using the SKR model exclusively fine-tuned on the unannotated training split of MMDocRAG, without any exposure to DocVQA or SciMMIR data during training. As detailed in the experimental results (see Appendix D.2), SKR models exhibit robust generalization, significantly outperforming advanced baseline RAG models.

A.4 CUB_200_2011

The CUB_200_2011 dataset (Wah et al., 2011) is a fine-grained image recognition benchmark containing 200 bird species, with a total of 5,994 training and 5,794 test samples. Each sample includes an image and its corresponding bounding-box annotation. To construct the imbalanced classification task, we selected hummingbirds (4 species, covering 2% of the data) as the rare positive class.

Consistent with our core methodology, we strictly adhere to an annotation-free regime during the training phase, using only the raw images from the training set without accessing any ground-truth labels or bounding boxes. Detailed performance results on this benchmark are presented in Appendix D.3.

B Evaluation Metrics

This section provides the formal definitions and calculation formulas for the metrics used to evaluate the performance of SKR across the three task types.

B.1 Information Retrieval (T_{IR})

For retrieval tasks, we focus on the model's ability to rank the ground-truth item as high as possible within a candidate pool. Inference latency is not reported for these tasks, as encoding times remain comparable between base and SKR-adapted models. Let Q be the set of test queries, and for each query $q \in Q$, let $rank_q$ denote the position of the most relevant document/image in the retrieved list.

1. Mean Reciprocal Rank (MRR): MRR calculates the average of the reciprocal ranks of the most relevant document for all queries:

$$\mathrm{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{6}$$

2. Recall@k (R@k): This metric measures the proportion of queries for which at least one relevant document/image is found within the top k retrieved results. Given our assumption that a single "golden truth" exists for each query in our dataset, this metric reflects the probability that the ground-truth item is captured within the top k results:

$$\text{Recall@k} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \mathbb{I}(rank_i \leqslant k) \tag{7}$$

where $\mathbb{I}(\cdot)$ is the indicator function. We report results using $k \in \{1, 3, 5, 10\}$ for our financial document datasets but using $k \in \{10, 15, 20\}$ for MMDocRAG to follow its metrics.

B.2 Efficient Region Localization (T_{Loc})

Region-localization performance is measured by the spatial overlap between the predicted bounding box and the ground truth, together with inference efficiency.

1. Intersection over Union (IoU): IoU quantifies the overlap between the predicted bounding box B_{p} and the ground-truth bounding box B_{gt}. It is calculated as the area of their intersection divided by the area of their union:

$$\mathrm{IoU} = \frac{\mathrm{Area}(B_{\mathrm{p}} \cap B_{\mathrm{gt}})}{\mathrm{Area}(B_{\mathrm{p}} \cup B_{\mathrm{gt}})} \tag{8}$$

We report the mean IoU across all test samples.

2. Inference Time (Time): We record the total running time required to process the test set and report it as minutes per 100 test cases. This metric highlights the efficiency gains of the task-specific head (E_T) compared to sequential token generation (E_{ntp}).

B.3 Imbalanced Classification (T_{IC})

For imbalanced classification, we evaluate both threshold-dependent and threshold-independent metrics.

1. Accuracy (Acc): The ratio of correctly predicted samples (both normal and anomalous) to the total number of samples:

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{9}$$

where $\mathrm{TP}, \mathrm{TN}, \mathrm{FP}, \mathrm{FN}$ represent true positives, true negatives, false positives, and false negatives, respectively.

2. AUROC: The Area Under the Receiver Operating Characteristic curve. It plots the True Positive Rate ($\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$) against the False Positive Rate ($\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}$) at various threshold settings. It measures the probability that a randomly chosen positive instance is ranked higher than a randomly chosen negative one.

3. AUPRC: The Area Under the Precision-Recall Curve. Precision ($P = \frac{\text{TP}}{\text{TP} + \text{FP}}$) is plotted against Recall ($R = \frac{\text{TP}}{\text{TP} + \text{FN}}$). AUPRC is more informative for the highly imbalanced classification datasets used here because it does not reward high numbers of True Negatives.

C Implementation Details

We introduce details of the self-annotation process, the fine-tuning settings, baseline settings, and various hyper-parameters for different tasks and different LLMs in this section.

C.1 Self-Annotation for Information Retrieval (T_{IR})

The self-annotation process includes using the LLM to generate text relevant to the input document or image. In this framework, we prompt the LLM to generate a specific query that acts as a semantic anchor. The prompt for text-to-text retrieval is:

Prompt for Text-to-text Retrieval Self-Annotation

```
Document:
<doc>
Based on the above document, generate one question that can be
answered using the information provided.
```

The prompt for text-to-image retrieval is:

Prompt for Text-to-image Retrieval Self-Annotation

```
<image title information>
Based on the above image, generate one question that can be answered
using the information provided.
```

Since the required annotations are direct, input-related textual queries, no additional post-processing is required for this task.

C.2 Self-Annotation for Region Localization (T_{Loc})

Due to the significant variance in the native capabilities of different models regarding T_{Loc}, we use three distinct self-annotation strategies: Direct Ask, Fuzzy Location, and the OCR-supported method.

C.2.1 Direct Ask

This method is applied to Qwen2-VL, as the base model possesses inherent grounding capabilities and can output formalized coordinates. We evaluate this model's potential using both one-turn and multi-turn prompting strategies. The one-turn prompt is:

Prompt for One-turn Region Localization

```
You are given a screenshot of a document page containing a target
object:  either a chart, table, or figure.  Your task is to identify
the bounding box of the whole object, including its title, and
the text underlying the source of this object.  Output normalized
coordinates of the bounding box relative to the image size.
```

The multi-turn prompts decompose the task into identifying the main object, its caption, and its source individually:

Prompts for Multi-turn Region Localization

```
You are given a screenshot of a document page containing a target
object:  either a chart, table, or figure.  Your task is to identify
the bounding box of the whole object.  Output normalized coordinates
of the bounding box relative to the image size.

You are given a screenshot of a document page containing a target
object:  either a chart, table, or figure.  Your task is to identify
the bounding box of this object's caption.  Output normalized
coordinates of the bounding box relative to the image size.

You are given a screenshot of a document page containing a target
object:  either a chart, table, or figure.  Your task is to identify
the bounding box of the text underlying the source of this object.
Output normalized coordinates of the bounding box relative to the
image size.
```

All outputs will be processed as four numerical numbers to represent the bounding boxes. For one-turn prompts, the output is used directly as the self-annotation. For multi-turn prompts, we calculate the minimum bounding rectangle that encompasses all three predicted boxes to form the final annotation.

C.2.2 Fuzzy Location

Since LLaVA-1.6 and Phi-3-VL lack native grounding abilities, we implement a bisection-based post-processing strategy to convert semantic understanding into spatial coordinates. We first prompt the model to describe the object:

Prompt for Object Description

```
You are given a screenshot of a document page which may contain a
target object:  either a chart, table, or figure.  Briefly describe
this object in no more than five words.
```

We apply this query to the original image and store the description. We then iteratively crop the image (e.g., moving the bottom boundary to half the image height) and re-query the model. If the semantic similarity between the new and original description exceeds 60%, we assume the object remains largely within the crop and continue the bisection. If similarity drops, we backtrack. This process repeats for all four boundaries until the boundary shift is less than 30 pixels, resulting in a "fuzzy" bounding box.

C.2.3 OCR-supported Method

To enhance annotation accuracy for weaker models, we invoke Tesseract's traditional non-neural OCR engine locally through `pytesseract`. This method provides coordinate support while remaining computationally much lighter than model-based approaches. We prompt the model to extract surrounding text like:

Prompt for Surrounding Text Extraction

```
You are given a screenshot of a document page containing a target
object:  either a chart, table, or figure.  Output the title of this
object.
The title of this object is:
```

We then use the OCR engine to locate the text on the page. By combining the bounding boxes of the extracted text anchors, we derive the smallest rectangle that covers the target area as self-annotation.

C.3 Self-Annotation for Imbalanced Classification (T_{IC})

For imbalanced classification, we provide the model with an image–query pair and use the following prompt:

Prompt for Imbalanced Classification

```
Based on the image above, determine whether the following question can
be answered using the information provided in the image.  Respond with
'yes' or 'no'.
Question:  <question>.
```

The textual responses are subsequently mapped to binary class labels (0 or 1) to form the training set.

C.4 Fine-tuning Details

The fine-tuning process uses task-specific output layers and specialized objective functions. For T_{IR}, we bypass the base model's token prediction layer and use the hidden state of the final token from the last layer as the dense representation. The model is fine-tuned using a standard contrastive learning objective. For T_{Loc}, we attach a regression head, a three-layer MLP, to the final hidden state of LLMs. The hidden size of the regression head is 1,024, and the final layer outputs 4 dimensions followed by a Sigmoid function to normalize coordinates between 0 and 1. We use L_1 loss for fine-tuning. For T_{IC}, we add a classification head (three-layer MLP, hidden size 1,024). The output is a single logit used for binary classification, optimized via binary cross-entropy loss.

All training sets are split into training and validation data at a 9:1 ratio. We use LoRA to adapt the models, adding trainable parameters to the query, key, value, and output projections (W_q, W_k, W_v, W_o) of all attention layers. The rank (r) is set to 8 and lora alpha (α) to 16. The total trainable parameters account for less than 0.5% of the total model size. The training configurations are tailored to the specific requirements of each task. For T_{IR}, the model is trained for 5 epochs with a batch size of 4, while T_{Loc} utilizes a batch size of 4 over 20 epochs. In the case of T_{IC}, the training duration is extended to 50 epochs with a batch size of 4. Across all three tasks, the models are optimized using the Adam optimizer (Kingma & Ba, 2014) with a consistent learning rate of 1×10^{-4}. Fine-tuning was conducted using four NVIDIA A100 GPUs. The process can be successfully executed on two GPUs if necessary.

D Additional Experimental Results

D.1 Text-to-text Retrieval Results of Llama-3.1 on Financial Document Dataset

Table 5 presents the performance of Llama-3.1. Consistent with Table 1, the SKR-adapted model shows substantial improvements and outperforms the included published embedding baselines on this dataset.

Table 5: Performance comparison of Llama3.1's text-to-text IR on the financial document dataset. SKR's results are highlighted in bold. Across all models and modalities, SKR consistently achieves the best performance.

Model		MRR	R@1	R@3	R@10
Text-to-text Retrieval					
Llama3.1	Base	0.196	0.136	0.214	0.330
	E5-V	0.306	0.240	0.318	0.426
	SFT	0.197	0.140	0.210	0.328
	SKR	**0.669**	**0.566**	**0.742**	**0.848**

D.2 TEXT-TO-IMAGE IR RESULTS ON MMDOCRAG, DOCVQA-2020, AND SCIMMIR

Table 6 presents the performance of SKR on the open-source MMDocRAG benchmark. To ensure a fair comparison, we use the reported results of the four top-performing models in text-to-image retrieval from their original publications. Our experimental setup strictly adheres to their established evaluation protocol and metrics. These baselines are leading retrieval models, including DSE (Ma et al., 2024), built on the Phi-3-Vision architecture, and ColQwen (Faysse et al., 2025), which uses the Qwen2-VL backbone.

The results show that all three SKR models outperform the reported specialized baselines under the official evaluation protocol. Relative to the strongest reported baseline, ColQwen, the SKR models improve Recall@10 by 12.6 to 16.9 percentage points.

Table 6: Performance comparison on MMDocRAG. SKR results are highlighted in bold. DSE and ColPali/ColQwen numbers are taken from the original benchmark publication rather than reproduced in-house.

Model		R@10	R@15	R@20
$\text{DSE}_{wiki-ss}$		0.671	0.753	0.799
$\text{DSE}_{\text{docmatix}}$		0.657	0.750	0.782
ColPali		0.682	0.775	0.812
ColQwen		0.708	0.792	0.843
LLaVA-1.6	Base	0.527	0.634	0.705
	SKR	**0.834**	**0.886**	**0.920**
Phi-3-VL	Base	0.607	0.703	0.768
	SKR	**0.842**	**0.891**	**0.923**
Qwen2-VL	Base	0.578	0.687	0.749
	SKR	**0.877**	**0.916**	**0.944**

Furthermore, Table 7 highlights the results of Qwen-SKR (fine-tuned exclusively on MMDocRAG) when evaluated on the DocVQA-2020 and SciMMIR test sets. Despite having no exposure to these datasets during the SKR process, our model significantly outperforms the Nomic baseline. This demonstrates SKR's exceptional cross-dataset generalization for similar tasks.

The fact that SKR can attain these results using self-generated query–image pairs shows that external retrieval pairs are not necessary for this particular adaptation setting. By transitioning to a task-optimized expression mechanism, SKR allows the model to use its internal knowledge far more effectively in specific tasks.

Table 7: Performance comparison on the DocVQA-2020 and SciMMIR datasets. SKR's results are highlighted in bold.

Model	**MRR**	**R@1**	**R@3**	**R@5**	**R@10**
DocVQA-2020					
Nomic	0.848	0.754	0.931	0.970	0.994
Qwen-SKR	**0.931**	**0.882**	**0.976**	**0.993**	**0.999**
SciMMIR					
Nomic	0.336	0.283	0.440	0.491	0.531
Qwen-SKR	**0.562**	**0.467**	**0.622**	**0.675**	**0.732**

D.3 LOCALIZATION AND IMBALANCED CLASSIFICATION RESULTS ON CUB_200_2011

Tables 8 and 9 present SKR performance on the open-source CUB_200_2011 benchmark for region localization and imbalanced classification. For T_{Loc}, IoU increases for all three models, with the runtime being reduced by at least 88%. This pattern broadly mirrors the efficiency gains on the financial dataset.

For T_{IC}, accuracy changes are small under extreme class imbalance, while AUROC and AUPRC provide more informative measures of ranking quality. Both metrics improve significantly for all three models.

These public-data results provide an additional check that the main trends are not restricted to the financial dataset, significantly improving performance while reducing runtime.

Table 8: Performance comparison on T_{Loc}. Results for the SKR method are highlighted in bold. Runtime is reported in minutes per 100 test cases.

Model	IoU	Time
Qwen Base	0.687	2.7
Qwen SKR	**0.692**	**0.3**
LLaVA Base	0.275	20.2
LLaVA SKR	**0.311**	**0.4**
Phi3 Base	0.237	19.3
Phi3 SKR	**0.266**	**0.4**

Table 9: Performance comparison on T_{IC}. SKR results are highlighted in bold. Runtime is reported in minutes per 100 test cases.

Model	Acc ↑	AUROC	AUPRC	Time
Qwen Base	0.985	0.857	0.471	0.8
Qwen SKR	**0.986**	**0.870**	**0.545**	**0.3**
LLaVA Base	0.965	0.840	0.246	2.2
LLaVA SKR	**0.966**	**0.918**	**0.442**	**0.3**
Phi3 Base	0.971	0.863	0.300	2.7
Phi3 SKR	**0.980**	**0.882**	**0.476**	**0.3**

E Impact of Model Scaling

To investigate the relationship between model scale and the effectiveness of SKR, we evaluate the Gemma-3 model family Team et al. (2025) across three parameter scales: 4b, 12b, and 27b. We specifically use the Gemma-3 family for this study rather than the models evaluated in our primary experiments to ensure a controlled comparison. While the previously evaluated models (e.g., Qwen, Phi, LLaVA) exhibit architectural and generational variances that could confound results, the Gemma-3 family provides a homogeneous environment where model size is the primary variable. This allows us to more rigorously isolate the impact of parameter scale on the SKR process.

The results in Table 10 examine how model size affects two observable factors: the quality of generated retrieval targets and the capacity of the adapted student. In addition to standard SKR, we include two cross-model supervision variants: (1) 27b-ft, where the 4b model is fine-tuned using targets from the 27b model, and (2) 4b-ft, where the 27b model is fine-tuned using targets from the 4b model.

Our results indicate that model size positively correlates with both the base model knowledge level and the performance of SKR-adapted models. The 12B and 27B versions outperform the 4B model, suggesting that larger architectures are more adept at internalizing and re-expressing information. However, this benefit exhibits asymptotic behavior; the performance gap between the 12B and 27B models is marginal, indicating a potential upper limit for the SKR process on this specific T_{IR} task.

The performance gains observed in larger models can be attributed to two distinct factors: annotation quality (the quality of self-generated annotations) and expressive capacity (the model's task ability to be adapted). The cross-model supervision experiments provide a clear decoupling of these factors:

Table 10: Performance comparison on T_{IR} for Gemma-3 models. SKR's results are highlighted in bold. The lines "27b-ft" and "4b-ft" denote models fine-tuned using annotations generated by the Gemma-3-27b-it and Gemma-3-4b-it models, respectively.

Model		MRR	R@1	R@3	R@5	R@10
Gemma-3 4b-it	Base	0.003	0.000	0.000	0.000	0.000
	E5-V	0.222	0.165	0.250	0.275	0.330
	SKR	**0.415**	**0.345**	**0.415**	**0.485**	**0.590**
	27b-ft	0.501	0.410	0.540	0.595	0.660
Gemma-3 12b-it	Base	0.025	0.010	0.020	0.025	0.045
	E5-V	0.293	0.230	0.305	0.355	0.425
	SKR	**0.526**	**0.420**	**0.580**	**0.635**	**0.736**
Gemma-3 27b-it	Base	0.051	0.035	0.050	0.070	0.080
	E5-V	0.047	0.025	0.060	0.065	0.080
	SKR	**0.531**	**0.440**	**0.580**	**0.625**	**0.715**
	4b-ft	0.472	0.375	0.535	0.580	0.635

The 4b model's performance increases significantly when fine-tuned on 27b-generated labels (27b-ft) compared to its own self-annotations. Conversely, the 27b model's performance drops when using 4b-generated labels (4b-ft). This confirms that larger models possess superior task-relevant knowledge, leading to higher-quality self-annotation.

Even when provided with high-quality 27b annotations, the 4b model still fails to match the performance of the 12b or 27b SKR-adapted models. Furthermore, the 27b model fine-tuned on 4b annotations (4b-ft) still outperforms the 4b model's best SKR results. This suggests that larger models have a higher architectural ceiling for re-expression, a more potential latent space that responds better to task-specific adaptation.

In summary, model size affects both the model's task-relevant knowledge and its potential for re-expression. Larger models not only generate superior self-annotations but also offer more potential for task-specific improvement. However, these gains do not scale indefinitely. Our findings suggest that we should calibrate model size to task difficulty, selecting the most efficient architecture that meets the required performance threshold.

F Ablation Study on LoRA Adapters

The SKR framework is designed as a knowledge re-expression process rather than a mere output formatting alignment for specific tasks. The LoRA adapters are essential to harmonize the model's internal parametric knowledge with the newly adapted output mechanism. To validate this design, we conducted ablation experiments comparing the full SKR framework against a "head-only" baseline. In the head-only configuration, we implement the task-specific head and loss functions but keep the backbone parameters frozen, bypassing the parameter adaptation phase.

Tables 11, 12, and 13 summarize these comparative results across information retrieval, region localization, and imbalanced classification.

The results clearly demonstrate that modifying the output head alone is insufficient for effective task adaptation. In T_{IR}, the R@10 gain achieved by SKR (0.299) is over 300% greater than that of the head-only baseline (0.074). In T_{Loc}, the head-only approach significantly degrades performance, with IoU dropping to 0.491, whereas SKR maintains and slightly improves upon the base model's accuracy while dramatically increasing efficiency. In T_{IC}, while the head-only model shows a marginal increase in AUPRC (0.021), it remains substantially lower than the improvement provided by SKR (0.074).

Collectively, these findings show that the gains are not a trivial byproduct of attaching a task head. The additional improvement from full SKR is consistent with the need to align models' hidden states with the task-native expression mechanism.

Table 11: Performance comparison on T_{IR} on the MMDocRAG dataset.

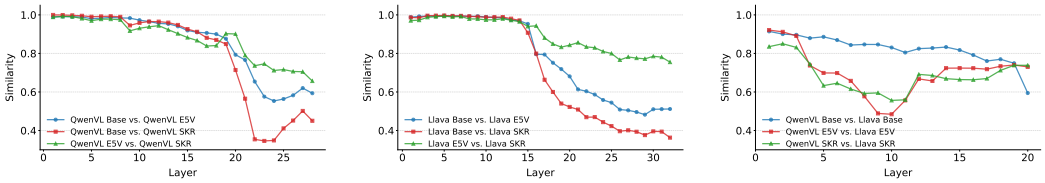
Model	R@10	R@15	R@20
Qwen Base	0.578	0.687	0.749
Qwen Head-only	0.652	0.735	0.797
Qwen SKR	**0.877**	**0.916**	**0.944**

Table 12: Performance comparison on T_{Loc} on the CUB dataset.

Model	IoU
Qwen Base	0.687
Qwen Head-only	0.491
Qwen SKR	**0.692**

Table 13: Performance comparison on T_{IC} on the CUB dataset.

Model	Acc	AUROC	AUPRC
Qwen Base	0.985	0.857	0.471
Qwen Head-only	0.980	0.856	0.492
Qwen SKR	**0.986**	**0.870**	**0.545**



(a) Layer-wise CKA similarity across methods for Qwen2-VL.

(b) Layer-wise CKA similarity across methods for LLaVA-1.6.

(c) Layer-wise CKA similarity between Qwen2-VL and LLaVA-1.6 under the same method.

Figure 4: Layer-wise representation similarity (CKA) under different settings.

G CKA Analysis

We analyze representation similarity across layers, methods, and architectures to gain deeper insights into how task-specific capabilities emerge within the model's latent space.

We use Centered Kernel Alignment (CKA; Kornblith et al. 2019) to quantify these similarities. CKA measures the alignment between representations in a kernel space, providing a robust metric that is invariant to isotropic scaling and orthogonal transformations. This method is widely adopted for studying the internal representational characteristics of LLMs (Zhao et al., 2024).

Figures 4(a) and 4(b) illustrate the layer-wise CKA similarity for Qwen2-VL and LLaVA-1.6 under different adaptation methods. In both architectures, hidden states remain highly similar in the early layers but diverge substantially in deeper layers. The SKR-adapted models, which achieve the highest performance, also show the lowest similarity to the corresponding base models after the midpoint layers. Meanwhile, the similarity between E5-V and SKR (green lines) remains relatively high compared with their similarity to the Base model, suggesting that different task-adaptation methods produce partially shared changes toward the task-specific objective. The sharp decline in similarity near the output layers indicates that SKR is accompanied by substantial changes in the hidden states supporting the task-native output mechanism.

Figure 4(c) further compares the two base architectures, Qwen2-VL and LLaVA-1.6. Because they differ in depth, we align their last 20 layers for comparison. Even when both models are adapted to the same task using SKR, their CKA similarity remains below 0.8. This shows that applying the same task objective does not homogenize the hidden states of different architectures. Instead, architecture-specific hidden-state structures remain after adaptation, while both models are aligned with the same task-native objective.