Generation-Augmented Generation: A Plug-and-Play Framework for Private Knowledge Injection in Large Language Models

Anonymous ACL submission

Abstract

In domains such as biomedicine, materials, and finance, high-stakes deployment of large language models (LLMs) requires injecting private, domain-specific knowledge that is proprietary, fast-evolving, and under-represented in public pretraining. However, the two dominant paradigms for private knowledge injection each have pronounced drawbacks: fine-tuning is expensive to iterate, and continual updates risk catastrophic forgetting and general-capability regression; retrieval-augmented generation (RAG) keeps the base model intact but is brittle in specialized private corpora due to chunk-induced evidence fragmentation, retrieval drift, and long-context pressure that yields query-dependent prompt inflation. Inspired by how multimodal LLMs align heterogeneous modalities into a shared semantic space, we propose **Generation-Augmented Generation (GAG)**, which treats private expertise as an additional expert modality and injects it via a compact, representation-level interface aligned to the frozen base model, avoiding prompt-time evidence serialization while enabling plug-and-play specialization and scalable multi-domain composition with reliable selective activation. Across two private scientific QA benchmarks (immunology adjuvant and catalytic materials) and mixed-domain evaluations, GAG improves specialist performance over strong RAG baselines by **15.34%** and **14.86%** on the two benchmarks, respectively, while maintaining performance on six open general benchmarks and enabling near-oracle selective activation for scalable multi-domain deployment.

1 Introduction

In recent years, large language models (LLMs) have demonstrated strong capabilities across a wide range of natural language processing tasks, including text understanding, generation, and instruction following (Grattafiori et al., 2024; Yang et al., 2025; Liu et al., 2024a; Guo et al., 2025). Pretrained on vast corpora of general text data, LLMs have profoundly impacted various aspects of daily life and professional environments. However, despite these impressive general capabilities, enabling LLMs to perform optimally in private domains remains a significant challenge. In private-domain deployments such as biomedicine, materials, and finance (Bao et al., 2023; Chen et al., 2023b,a), reliable performance often requires incorporating domain-specific knowledge beyond open-domain pretraining, where expert terminology and conventions are critical for accurate and dependable outputs.

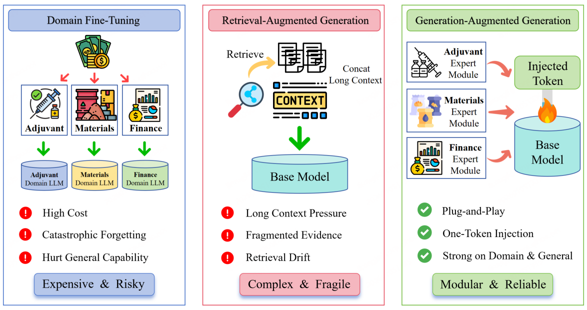

Figure 1: **Three paradigms for private knowledge injection.** Fine-tuning is expensive and risky; RAG is complex and fragile due to retrieval and long-context pressure. **GAG** injects private expertise through a constant-budget, modular interface with selective activation.

Two dominant paradigms are commonly used to inject private knowledge into LLMs. **(i) Domain fine-tuning** can internalize domain knowledge, but it is costly to iterate, requires careful validation, and risks general-capability regression and catastrophic forgetting under continual updates (Gururangan et al., 2020; Hu et al., 2022; Dettmers et al., 2023). **(ii) Retrieval-augmented generation (RAG)** preserves the base model by retrieving textual evidence at inference time (Lewis et al., 2020; Guu et al., 2020; Izacard and Grave, 2021; Izacard et al., 2023). However, in private domains RAG is often brittle: evidence is fragmented by chunking, retrieval can drift or miss crucial context, and even relevant passages must compete for limited context budget and are unevenly utilized by long-context LLMs (Liu et al., 2024b). Moreover, because evidence must be serialized into the prompt, RAG induces query-dependent context expansion, making inference behavior less predictable as the private corpus scales. Figure 1 shows these trade-offs and positions **Generation-Augmented Generation (GAG)** as a constant-budget, modular alternative to both fine-tuning and retrieval-based injection.

In this work, we reformulate private knowledge injection from a multimodal perspective, conceptualizing such knowledge not as textual snippets but as an auxiliary modality. This modality can be aligned and fused into the general LLM via parameter-efficient interfaces (Alayrac et al., 2022; Li et al., 2023; Liu et al., 2023; Huang et al., 2023). To overcome the limitations of both fine-tuning and RAG, we introduce **GAG**: a retrieval-free, plug-and-play framework that performs private knowledge injection through a constant-budget interface, without updating the parameters of the frozen base model. The framework adopts a decoupled architecture that partitions the system into a general-purpose base model and a set of lightweight domain-specific expert modules. A routing mechanism dynamically selects between a general route and domain-specific routes. This design transforms private-knowledge injection into a constant-budget operation by injecting a single continuous token into the frozen base model, avoiding both base-model fine-tuning and retrieval-time evidence serialization, thereby enabling efficient and scalable integration of knowledge across multiple domains while preserving general-domain capability.

Contributions. (1) We introduce GAG, a retrieval-free and plug-and-play framework for private knowledge injection into a frozen base model via a single-token continuous interface. (2) We propose a prototype-based global router that enables reliable selective activation and incremental multi-domain expansion without router training. (3) We validate GAG on two private scientific domains and mixed-domain settings, demonstrating substantial specialist improvements while maintaining general-domain performance.

2 Related Work

Fine-tuning-based knowledge injection. Parametric adaptation injects domain knowledge via continued pretraining or supervised fine-tuning on domain data (Gururangan et al., 2020). A key challenge is catastrophic forgetting and general-capability regression under continual updates, unless continual-learning controls are applied (Kirkpatrick et al., 2017; Li and Hoiem, 2017). To reduce cost, parameter-efficient fine-tuning (PEFT) updates only small parameter subsets (e.g., adapters, prefix/prompt tuning, sparse updates) (Houlsby et al., 2019; Li and Liang, 2021; Lester et al., 2021; Zaken et al., 2022). Low-rank and quantization-aware PEFT further improve efficiency and memory (Hu et al., 2022; Zhang et al., 2023; Mao et al., 2022; Pfeiffer et al., 2021; Dettmers et al., 2023; Lialin et al., 2023). However, they still require iterative training/re-validation and maintain evolving domain parameters, which conflicts with deployments requiring a strictly frozen base model for governance and regression control. This motivates modular knowledge injection mechanisms that preserve a frozen base model while supporting plug-and-play domain expansion.

Retrieval-augmented knowledge injection. Retrieval-augmented generation (RAG) injects external knowledge by retrieving evidence from a corpus and conditioning an LLM on the retrieved text, and it is widely adopted for knowledge-intensive QA (Lewis et al., 2020; Guu et al., 2020; Izacard and Grave, 2021). A large body of work improves the retrieve-then-read pipeline via stronger dense retrieval and late-interaction matching, better training objectives, and more effective reader-side fusion (Karpukhin et al., 2020; Xiong et al., 2020; Khattab and Zaharia, 2020). Other lines integrate retrieval more tightly into training/inference, or plug retrieval into pretrained LMs to improve factual grounding and sample efficiency (Izacard et al., 2023; Borgeaud et al., 2022; Shi et al., 2024; Khandelwal et al., 2019). More recently, LM-driven generation or verification signals have been explored to improve retrieval robustness and attribution faithfulness (Gao et al., 2023b,a; Asai et al., 2024). Despite these advances, RAG can be particularly challenging in private, fast-evolving domains: evidence is fragmented by chunking, top-k retrieval does not guarantee complete coverage, and long-context LMs may under-utilize or misinterpret relevant spans when multiple passages compete within a finite context budget (Liu et al., 2024b; Bai et al., 2024). Context-compression methods (e.g., one-token interfaces) reduce prompt overhead, but remain retrieval-dependent and thus still hinge on retrieval coverage and indexing quality (Cheng et al., 2024). Meanwhile, auxiliary-model-based knowledge transfer can inject domain signals through prompt-time mediation (Li et al., 2025), but still relies on textual handoff and typically requires non-trivial organization of training data and prompts, while remaining subject to context-budget pressure, making modular multi-domain scaling less straightforward. In contrast, our work targets a retrieval-free, constant-budget injection interface under a frozen base model, avoiding prompt-time evidence serialization while enabling predictable inference behavior, requiring only minimal per-domain data preparation, and supporting plug-and-play multi-domain composition with reliable selective activation via routing.

3 Problem Formulation

We consider question answering with a frozen base model. Let $p_\theta(y \mid x)$ denote the conditional distribution induced by a pretrained model with parameters θ, where x is a user query and y is the target answer. After deployment, θ is not allowed to be updated.

Multi-domain private knowledge. Queries are drawn from a mixture of one general distribution $\mathcal{D}_0$ and N private-domain distributions $\{\mathcal{D}_i\}_{i=1}^N$. For each domain i, samples $(x, y) \sim \mathcal{D}_i$ reflect domain-specific knowledge needs (e.g., enterprise or vertical expertise) that are not reliably covered by public pretraining. We assume each private domain i is associated with a private knowledge source $\mathcal{K}_i$ (e.g., internal documents or curated resources), but we do not retrain the base model on $\mathcal{K}_i$.

Knowledge injection as conditional generation with side information. Our objective is to enable the frozen base model to answer domain-specific queries without modifying its parameters. To this end, we allow the system to condition the base model on an auxiliary injected signal z derived from $(x, \mathcal{K}_i)$:

$$z = \mathcal{A}_i(x, \mathcal{K}_i), \qquad \hat{y} \sim p_\theta(y \mid x, z), \quad (1)$$

where $\mathcal{A}_i$ is an abstract domain-specific injection mechanism. Eq. (1) abstracts away the form of

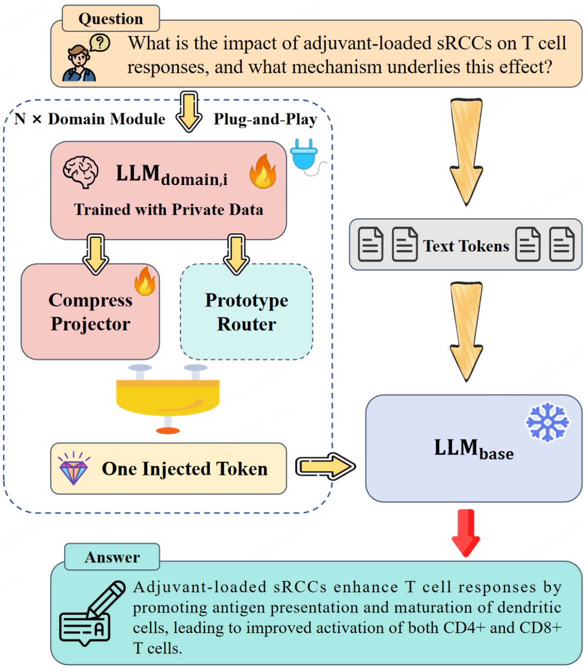

Figure 2: **GAG overview.** A training-free prototype router selects either the general route or one of N plug-and-play domain modules. Each module derives an expert readout from $\text{LLM}_{\text{domain},i}$ and projects it into LLM_{base}'s embedding space as a single continuous injected token, enabling constant-budget, retrieval-free knowledge injection under a frozen base model.

z and its integration, requiring only an external modular interface that can influence generation.

Objective and constraints. Our goal is to improve private-domain QA quality while preserving the base model's general capability. Let $\mathcal{L}(\hat{y}, y)$ be a task loss (or an evaluation-aligned surrogate). We seek injection mechanisms $\{\mathcal{A}_i\}_{i=1}^N$ such that:

$$\begin{aligned} \min_{\{\mathcal{A}_i\}_{i=1}^N} \quad & \sum_{i=1}^{N} \mathbb{E}_{(x,y)\sim\mathcal{D}_i}[\mathcal{L}(\hat{y}_i(x), y)] \\ \text{s.t.} \quad & \mathbb{E}_{(x,y)\sim\mathcal{D}_0}[\mathcal{L}(\hat{y}_0(x), y)] \le R_0 + \epsilon. \end{aligned} \quad (2)$$

where R_0 denotes the baseline risk of the frozen base model on $\mathcal{D}_0$ (without injection) and ϵ is an allowable regression margin.

Plug-and-play domain expansion. We further require modular multi-domain expansion: when a new domain k arrives, the system should incorporate $\mathcal{A}_k$ without modifying the base-model parameters θ or previously deployed mechanisms $\{\mathcal{A}_i\}_{i<k}$. This captures the practical constraint that private knowledge evolves across domains while the base model must remain stable and reusable.

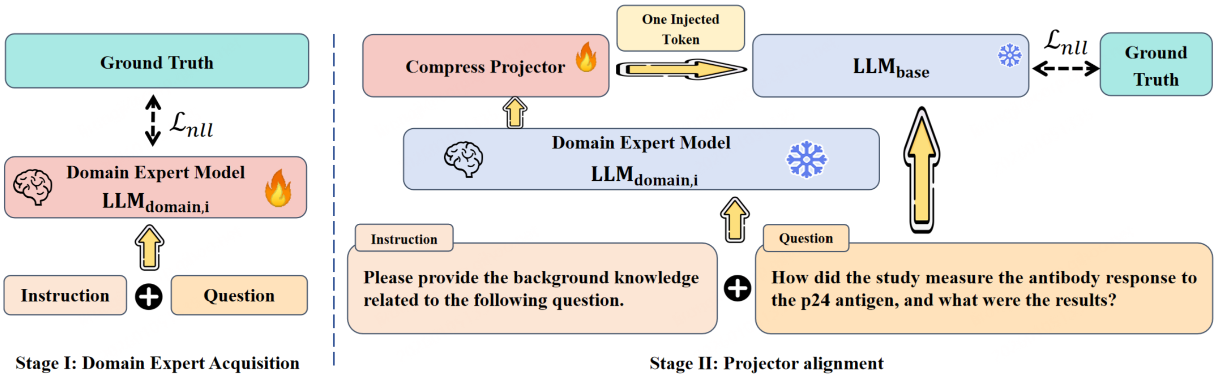

Figure 3: **GAG two-stage learning. Stage I** distills domain competence into LLM$_{\text{domain},i}$'s internal representations. **Stage II** learns a lightweight projector Π_i that maps the expert readout $\mathbf{k}_i(x)$ into LLM$_{\text{base}}$'s embedding space, yielding a single continuous injected token for constant-budget, plug-and-play knowledge injection.

4 Method

We propose **GAG**, which injects private knowledge into a strictly frozen base model by attaching lightweight domain modules that synthesize holistic domain background and align it via a learned one-token interface, with a prototype router activating the appropriate route on demand. Figure 2 depicts the end-to-end routed pipeline.

4.1 GAG Architecture

Let LLM$_{\text{base}}$ be a frozen base model with parameters θ and hidden size d_1. Given a query x, GAG first predicts a route index

$$\hat{i} = r(x) \in \{0, 1, \dots, N\}, \tag{3}$$

then produces an answer by conditioning LLM$_{\text{base}}$ on a route-specific side signal $z_{\hat{i}}(x) \in \mathbb{R}^{d_1}$:

$$\hat{y} \sim p_\theta\big(y \mid x, z_{\hat{i}}(x)\big), \qquad z_0(x) \triangleq \text{NULL}. \tag{4}$$

Routes $i \geq 1$ correspond to domain expert modules (Section 4.2); $r(x)$ is realized by PPR (Section 4.3).

4.2 Generation Augmentation for Each Specialist Domain

We instantiate the injected signal $z_i(x)$ as a compact representation-level summary of domain expertise, obtained from a domain expert model and aligned to LLM$_{\text{base}}$'s input embedding space via a lightweight projector.

Expert background synthesis and semantic readout. For each domain i, a lightweight causal LM LLM$_{\text{domain},i}$ (parameters ϕ_i, hidden size d_2) generates a background sequence:

$$b_{1:T} \sim p_{\phi_i}(b \mid x). \tag{5}$$

Let $\mathbf{h}_t^{(\ell)} \in \mathbb{R}^{d_2}$ denote the hidden state at step t and layer ℓ. We compress the generated background into a single expert vector via a late-layer readout:

$$\mathbf{k}_i(x) = \mathbf{h}_T^{(\ell^\star)} \in \mathbb{R}^{d_2}, \qquad \ell^\star \in \{1, \dots, L_2\}, \tag{6}$$

where L_2 is the number of transformer layers in LLM$_{\text{domain},i}$ and $\ell^\star$ is a readout layer (ablation in Section 7.1).

Token-geometry alignment and one-token injection. We learn a lightweight projector $\Pi_i : \mathbb{R}^{d_2} \to \mathbb{R}^{d_1}$ (parameters ψ_i) to map the expert vector into LLM$_{\text{base}}$'s embedding space:

$$z_i(x) = \Pi_i(\mathbf{k}_i(x)) \in \mathbb{R}^{d_1}. \tag{7}$$

To condition LLM$_{\text{base}}$, we fix an anchor slot at position a in the prompt template $s_{1:n}$ and substitute its input embedding. Let $\mathbf{E}_\theta(s_{1:n}) \in \mathbb{R}^{n \times d_1}$ denote LLM$_{\text{base}}$'s input embeddings; we form

$$\mathbf{E}_\theta^{(i)}(x) = \mathbf{E}_\theta(s_{1:n}) \text{ with } \mathbf{E}_\theta(s_a) \leftarrow z_i(x), \tag{8}$$

and decode with the frozen base model:

$$\hat{y} \sim p_\theta\big(y \mid \mathbf{E}_\theta^{(i)}(x)\big). \tag{9}$$

Eq. (8) defines a constant-budget knowledge interface (one injected token), related to continuous prompting (Li and Liang, 2021; Lester et al., 2021) but expert-generated and cross-model aligned.

Figure 3 illustrates the two-stage learning procedure.

Stage I: domain expert acquisition via QA adaptation. We first endow LLM$_{\text{domain},i}$ with domain expertise by adapting it on in-domain QA pairs. Concretely, given $(x, y) \sim \mathcal{D}_i$, we train

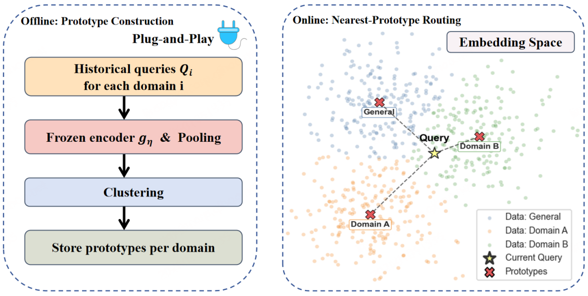

Figure 4: **Prototype Plug-and-Play Routing (PPR). Offline:** embed historical queries with a frozen encoder g_η and cluster them into per-domain prototype banks. **Online:** route each query by nearest-prototype matching to select the general path or a domain module, enabling training-free, scalable multi-domain expansion without updating the base model.

$\text{LLM}_{\text{domain},i}$ to model the domain-conditional answer distribution, thereby internalizing salient domain regularities that later serve as a strong expert prior for background synthesis. Formally, we optimize the standard autoregressive objective

$$\min_{\phi_i} \mathbb{E}_{(x,y)\sim\mathcal{D}_i}\Big[-\sum_{t=1}^{|y|}\log p_{\phi_i}(y_t \mid y_{<t}, x)\Big]. \quad (10)$$

After this adaptation, $\text{LLM}_{\text{domain},i}$ can be prompted to produce holistic domain background as an intermediate signal.

Stage II: projector alignment under a frozen base model. We then freeze θ and ϕ_i, and learn only the projector parameters ψ_i by maximizing the likelihood of the gold answer under injected decoding:

$$\min_{\psi_i} \mathbb{E}_{(x,y)\sim\mathcal{D}_i}\Big[-\sum_{t=1}^{|y|}\log p_{\theta}\big(y_t \mid y_{<t}, \mathbf{E}_\theta^{(i)}(x)\big)\Big]. \quad (11)$$

This objective directly optimizes the injection interface in LLM_{base}'s native representational space, enabling domain specialization without updating the base-model parameters.

Overall, in Stage I, we adopt the simplest training scheme to endow the lightweight $\text{LLM}_{\text{domain},i}$ with domain knowledge; in Stage II, we further enable it to produce background knowledge while aligning its outputs to the frozen LLM_{base}. This design is simple yet effective, and it naturally supports reusing domain models released by prior work.

4.3 Prototype Plug-and-Play Routing

To enable plug-and-play multi-domain expansion and reliable selective activation, we incorporate Prototype Plug-and-Play Routing (PPR) as a built-in component of GAG. PPR implements $r(x)$ in Eq. (3) via a training-free prototype-based decision rule, serving as a non-parametric alternative to learned routers in conditional computation / MoE systems (Lepikhin et al., 2020; Fedus et al., 2022).

Embeddings and prototypes. Let g_η be a frozen encoder and $\text{Pool}(\cdot)$ a fixed pooling operator. We embed and normalize:

$$\mathbf{e}(x) = \frac{\text{Pool}(g_\eta(x))}{\|\text{Pool}(g_\eta(x))\|_2} \in \mathbb{R}^d. \quad (12)$$

Offline, for each route i, we cluster historical query embeddings into C_i prototypes:

$$\begin{aligned}\mathbf{P}_i &= \{\mathbf{p}_{i,1}, \ldots, \mathbf{p}_{i,C_i}\} \\ &= \text{KMeans}(\{\mathbf{e}(x)\}_{x\in\mathcal{Q}_i};\, C_i), \qquad \|\mathbf{p}_{i,c}\|_2 = 1.\end{aligned} \quad (13)$$

Nearest-prototype routing. At inference time, PPR routes by nearest-prototype cosine similarity:

$$\begin{aligned} s_i(x) &= \max_c \ \mathbf{e}(x)^\top \mathbf{p}_{i,c}, \\ r(x) &= \arg\max_{i\in\{0,\ldots,N\}} s_i(x). \end{aligned} \quad (14)$$

Figure 4 visualizes the embedding space, domain prototypes, and the nearest-prototype decision boundary.

Modularity and incremental deployment. GAG supports domain expansion by construction: adding a new domain k only requires attaching $(\text{LLM}_{\text{domain},k}, \Pi_k)$ and computing prototypes $\mathbf{P}_k$, while keeping LLM_{base} and existing modules unchanged. This isolates domain updates from the base model, mitigating the iteration cost and regression risks typically associated with repeated fine-tuning.

5 Experimental Setup

5.1 Datasets and Metrics

We evaluate GAG on both **general-domain QA** and **specialist private-domain QA** to quantify whether modular knowledge injection improves domain expertise without compromising broad usability. For general QA, we follow prior work and report performance on six widely used open-domain benchmarks—FreebaseQA (Jiang et al., 2019), HotpotQA (Yang et al., 2018), Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), WebQuestions (Berant et al., 2013), and PopQA (Mallen et al., 2022)—using **Exact Match (EM)** with standard answer normalization, which provides a stringent measure of factual correctness under canonical string matching. To study domain knowledge injection, we focus on two specialist domains: **immunology adjuvant** and **catalytic materials**. Concretely, we treat (Anonymous, 2025b) and (Anonymous, 2025a) as the supervision sources for domain expert knowledge injection, and evaluate on their held-out test sets to quantify specialist QA quality. Because reference answers in these domains are often free-form and allow surface variation, we report **BERTScore** (Zhang et al., 2019) computed with SciBERT (Beltagy et al., 2019), a scientific-domain encoder that better aligns evaluation with semantic faithfulness in technical language. More detailed dataset statistics are provided in Appendix A.

5.2 Implementation Details

We instantiate the frozen base model $\mathrm{LLM}_{\mathrm{base}}$ with **Qwen3-8B** and domain expert models $\mathrm{LLM}_{\mathrm{domain},i}$ with **Qwen3-1.7B** (Yang et al., 2025). The projector Π_i is implemented as a lightweight two-layer MLP with a GELU nonlinearity. For routing, we use **Qwen3-1.7B** as a frozen query encoder and treat general as a peer route: all domains compete via maximum prototype cosine similarity, requiring neither router training nor threshold tuning. All experiments are run on 8×NVIDIA A100 GPUs with bfloat16 precision and FlashAttention-2 (Dao, 2024) when available; full training hyperparameters, routing configuration, inference and decoding settings, and prompt templates are provided in Appendix B and Appendix C.

5.3 Baselines

We compare against representative and competitive knowledge-injection baselines under a frozen base model constraint: **(i) Base-Model-Only**, where Qwen3-8B answers directly without any external knowledge; **(ii) RAG**, which builds domain corpora by parsing scientific papers with MinerU2.5 (Niu et al., 2025), retrieves top-k domain evidence via ColBERTv2 (Santhanam et al., 2022), and conditions the same base model on the retrieved text; **(iii) xRAG** (Cheng et al., 2024), which performs retrieval augmentation by retrieving the top-1 background passage from the corresponding domain knowledge base and compressing it under an extreme budget; and **(iv) Expert-Generated Context (EGC)**, where a domain expert model first generates explicit background text that is appended to the base model prompt (i.e., text-level knowledge transfer without a learned knowledge injection interface). Oracle routing is treated as a diagnostic upper bound and is reported in Section 6.1.

6 Experimental Results

6.1 Overall Performance

Table 1 reports overall performance on two specialist domains (Adjuvant, Materials) and six general-domain QA benchmarks under a strictly frozen Qwen3-8B base model. We omit RAG/xRAG on general-domain benchmarks because they serve as closed-book regression checks; adding open-domain retrieval would change the setting, while disabling retrieval makes RAG/xRAG identical to the base-model-only route. Overall, **GAG delivers the strongest specialist gains among all baselines while preserving general-domain capability.**

On specialist QA, **GAG delivers large and consistent gains** over both retrieval-based baselines and text-level expert transfer. RAG improves only marginally, and xRAG remains close to RAG even under a one-token budget, indicating that the dominant bottleneck is retrieval reliability rather than context length alone: chunk fragmentation, entity drift, and incomplete coverage mean that compress-

System	Specialist Domain		General Domain							Added Tokens
	Adjuvant	Materials	FreebaseQA	HotpotQA	Natural Questions	TriviaQA	WebQuestions	PopQA	Average	
Base-Model-Only	56.12	60.01	61.06	28.72	35.15	**54.36**	49.96	23.70	42.16	0
RAG	59.97(6.86%)	62.13(3.53%)				—				375.17
xRAG	59.12(5.35%)	62.24(3.72%)				—				1
EGC (Adjuvant)	64.07(14.17%)	58.60(-2.35%)	38.50(-36.95%)	18.24(-36.49%)	17.09(-51.38%)	34.54(-36.46%)	30.22(-39.51%)	14.36(-39.41%)	25.49(-39.54%)	148.48
EGC (Materials)	54.22(-3.39%)	66.47(10.76%)	38.59(-36.80%)	21.23(-26.08%)	18.85(-46.37%)	35.24(-35.17%)	33.22(-33.51%)	14.89(-37.17%)	27.00(-35.96%)	165.21
GAG (Gen+Adj)	69.16(23.24%)	—	**61.41(0.57%)**	28.46(-0.91%)	**35.59(1.25%)**	54.01(-0.64%)	49.96(0.00%)	24.41(3.00%)	42.31(0.36%)	1
GAG (Gen+Adj+Mat)	**69.17(23.25%)**	**71.36(18.91%)**	**61.41(0.57%)**	**29.16(1.53%)**	34.98(-0.48%)	53.57(-1.45%)	**50.31(0.70%)**	**24.67(4.09%)**	**42.35(0.45%)**	1

Table 1: **Overall performance across specialist and general domains.** Specialist-domain QA is evaluated on Adjuvant and Materials (BERTScore with SciBERT, ×100), and general-domain QA is evaluated with Exact Match (EM) on six open benchmarks. **Bold** indicates the best result in each column and underline indicates the second best. Numbers in parentheses denote relative improvements over the Base-Model-Only (Qwen3-8B) baseline. Added Tokens reports the average number of additional knowledge tokens injected into the base-model prompt.

ing retrieved text cannot fix missing or misaligned evidence. EGC exposes a complementary failure mode: although a matched expert can help, directly appending generated background is an inherently high-variance text-level intervention—its relevance and granularity are not guaranteed—so the added context can dilute or misguide the base model's focus under attention competition, limiting in-domain robustness. In contrast, GAG transfers domain competence through a representation-level interface—aligning an expert readout to the frozen base model's embedding geometry—yielding a compact, high-signal conditioning token that avoids prompt-time evidence competition and consistently improves specialist performance across domains. RAG results under different retrieval depths are reported in Appendix D.

Importantly, these specialist improvements do not come at the expense of general QA. **GAG maintains the general-domain average** relative to the base model, whereas EGC serves as a stress test showing that indiscriminate text-level injection can severely degrade open-domain performance. This contrast highlights the practical necessity of reliable selective activation. Finally, GAG operates with a **constant knowledge budget** yet closely approaches the oracle-route upper bounds on specialist domains (69.72 on Adjuvant and 71.53 on Materials under oracle domain routing), offering a deployment-friendly alternative to variable-length retrieval and text conditioning.

Router configuration	Active routes	Micro acc. (%)	Per-route acc. (%)		
			Gen	Adj	Mat
PPR (2 routes; base)	Gen + Adj	99.78	99.65	99.91	—
PPR (3 routes; incremental +Mat)	Gen + Adj + Mat	99.55	99.65	99.38	99.69

Table 2: **Plug-and-play routing accuracy of PPR.** Active routes list the domains with loaded prototype banks. Micro acc. is micro-averaged route-selection accuracy over all evaluated queries. Per-route acc. reports class-wise accuracy for each route (Gen/Adj/Mat).

6.2 Routing Accuracy of PPR

Reliable selective activation is a prerequisite for plug-and-play expert deployment, since misrouting can turn knowledge injection into harmful interference. Table 2 shows that **PPR achieves near-oracle routing under a fully frozen setup**: using a frozen Qwen3-1.7B encoder and nearest-prototype matching, PPR attains **99.78%** micro-averaged accuracy for Gen+Adj, and remains **99.55%** after incrementally adding Mat without modifying any existing routes. Per-route accuracy stays uniformly high, indicating that PPR provides a stable, non-parametric routing interface for scalable plug-and-play expert composition; additional routing results under broader incremental domain expansion are deferred to the appendix E.

$\mathrm{LLM}_{\mathbf{domain},i}$ **readout layer**	**BERTScore**↑	Δ
L_2-1	69.12	−0.60
L_2-2	69.20	−0.52
L_2-4 **(default)**	**69.72**	+0.00
L_2-6	69.36	−0.36
L_2-8	69.38	−0.34
L_2-12	69.14	−0.58
L_2-16	68.87	−0.85
L_2-20	66.69	−3.03
L_2-24	58.42	−11.30

Table 3: **Readout-layer ablation for one-token GAG injection.** We form the injected token by projecting the last-token hidden state from $\mathrm{LLM}_{\mathrm{domain},i}$ layer ℓ into the frozen base model $\mathrm{LLM}_{\mathrm{base}}$. We disable routing and always activate the Adjuvant module (oracle domain routing) to isolate the effect of readout depth. BERTScore↑ (SciBERT) is evaluated on Adjuvant; Δ is the difference relative to the best readout depth ($\ell = L_2 - 4$), which we use as default.

7 Analysis

7.1 Which $\mathrm{LLM}_{\mathbf{domain},i}$ layer yields the best one-token readout?

GAG relies on a single-vector readout $\mathbf{k}_i(x)$ from $\mathrm{LLM}_{\mathrm{domain},i}$ (Eq. 6), making the readout depth a key determinant of injection bandwidth and stability. We ablate the readout layer by extracting the last-token hidden state from $\ell \in \{L_2–1, L_2–2, L_2–4, \ldots, L_2-24\}$, projecting it into $\mathrm{LLM}_{\mathrm{base}}$, and injecting exactly one continuous token. All other factors (frozen $\mathrm{LLM}_{\mathrm{base}}$/$\mathrm{LLM}_{\mathrm{domain},i}$, prompts, decoding) are held constant.

Table 3 shows a sharp depth effect on Adjuvant: $\ell = L_2-4$ is optimal and a narrow late-layer band remains competitive, while substantially earlier layers collapse performance (e.g., L_2-20/L_2-24). This pattern suggests that effective one-token transfer prefers representations that are semantically consolidated yet not overly specialized to next-token prediction. Accordingly, we set $\ell^\star = L_2-4$ as the default readout layer in all experiments for robust plug-and-play deployment under a fully frozen base model.

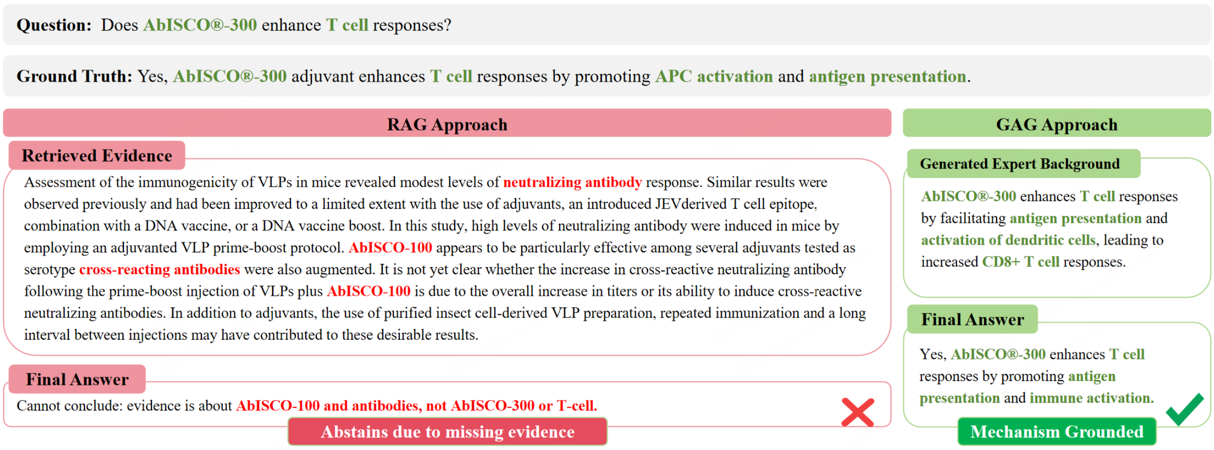

Figure 5: **Case study (RAG vs. GAG).** We contrast RAG's retrieved evidence and answer with GAG's injected-token route and answer for the same query and reference. The displayed "Generated Expert Background" is an analysis-only visualization of the injected signal (not exposed in deployment).

Variant	Stage I	Stage II	BERTScore↑
w/o Stage I	×	✓	57.14
w/o Stage II	✓	×	55.64
Full	✓	✓	**69.72**

Table 4: **Two-stage ablation of GAG (Adjuvant).** All variants are evaluated with oracle domain routing.

7.2 Are both Stage I and Stage II necessary?

We conduct a two-stage training ablation of GAG on Adjuvant to quantify the necessity of Stage I and Stage II (Table 4). **w/o Stage I** keeps the alignment objective but starts from a non-specialized $\mathrm{LLM}_{\mathrm{domain},i}$, reducing BERTScore to 57.14, which indicates that the injected representation lacks domain-sufficient signal. **w/o Stage II** preserves an adapted expert but replaces alignment with an untrained mapping, further degrading to 55.64, suggesting that $\mathrm{LLM}_{\mathrm{domain},i}$ representations are not directly usable in $\mathrm{LLM}_{\mathrm{base}}$'s embedding space. The full model reaches 69.72, confirming that both stages are complementary prerequisites for effective one-token injection.

7.3 Case study

Figure 5 illustrates retrieval brittleness in chunked private corpora: the query targets AbISCO-300 and a T-cell/APC mechanism, yet RAG's evidence is **entity-mismatched** (AbISCO-100) and dominated by humoral readouts, so the frozen base model cannot ground the requested mechanism and effectively abstains. GAG instead conditions the base model via a constant-budget injected token derived from the domain expert module, avoiding prompt-time evidence serialization and its coverage gaps. Importantly, the expert background text shown in the figure is only an analysis-time probe for interpretability: in the actual GAG pipeline, $\mathrm{LLM}_{\mathrm{domain},i}$ produces **no explicit text output** to the user and contributes solely a single continuous embedding. This case supports our claim that GAG mitigates retrieval fragmentation/mismatch while keeping the interface fixed and predictable under a frozen base model. In Appendix F, we include more interesting cases including error analysis.

8 Conclusion

We proposed **GAG**, a retrieval-free, plug-and-play framework for injecting private, domain-specific knowledge into a frozen base model via a constant-budget one-token interface aligned from a lightweight domain expert model's hidden representation. By moving from text-level evidence serialization to representation-level knowledge transfer, GAG directly targets key deployment bottlenecks of prevailing paradigms: it avoids RAG's brittleness under chunking, context-window pressure, and retrieval mismatch/noise, while sidestepping the iteration cost and general-capability regression risks that often accompany fine-tuning in continuously evolving private domains. Across two private scientific QA benchmarks and a mixed-domain setting, GAG yields strong specialist gains while preserving general QA, highlighting a practical route toward modular, scalable, and governance-friendly private-knowledge deployment in real-world LLM systems.

9 Limitations

While GAG demonstrates strong effectiveness for private knowledge injection, we note two limitations. First, our formulation assumes that each query is predominantly associated with a single domain. For genuinely cross-domain questions that require composing knowledge from multiple private domains, GAG typically injects knowledge from only one selected private domain, which may limit multi-domain knowledge fusion and lead to degraded performance on such mixed queries. In future work, probabilistic multi-domain joint injection may help mitigate this limitation. Second, because GAG injects knowledge via a single continuous token rather than verbatim evidence text, it may occasionally be less precise on exact surface-form numerics/units when the evaluation hinges on copying a specific number (as illustrated in Appendix F); in practice, lightweight post-hoc numeric/unit normalization or verification can complement the constant-budget interface without changing the core method.

References

Ankush Agarwal, Sakharam Gawade, Sachin Channabasavarajendra, and Pushpak Bhattacharyya. 2022. There is no big brother or small brother: knowledge infusion in language models for link prediction and question answering. In *Proceedings of the 19th International Conference on Natural Language Processing (ICON)*, pages 204–211.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, and 1 others. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Anonymous. 2025a. Catalystbench: A comprehensive multi-task benchmark for advancing language models in catalysis science. In *Submitted to The Fourteenth International Conference on Learning Representations*. Under review.

Anonymous. 2025b. An open-ended benchmark and formal framework for adjuvant research with MLLM. In *Submitted to The Fourteenth International Conference on Learning Representations*. Under review.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, and 1 others. 2024. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137.

Zhijie Bao, Wei Chen, Shengze Xiao, Kuang Ren, Jiaao Wu, Cheng Zhong, Jiajie Peng, Xuanjing Huang, and Zhongyu Wei. 2023. Disc-medllm: Bridging general large language models and real-world medical consultation. *arXiv preprint arXiv:2308.14346*.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. In *EMNLP*. Association for Computational Linguistics.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, and 1 others. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Wei Chen, Qiushi Wang, Zefei Long, Xianyin Zhang, Zhongtian Lu, Bingxuan Li, Siyuan Wang, Jiarong Xu, Xiang Bai, Xuanjing Huang, and 1 others. 2023a. Disc-finllm: A chinese financial large language model based on multiple experts fine-tuning. *arXiv preprint arXiv:2310.15205*.

Zi-Yi Chen, Fan-Kai Xie, Meng Wan, Yang Yuan, Miao Liu, Zong-Guo Wang, Sheng Meng, and Yan-Gang Wang. 2023b. Matchat: A large language model and application service platform for materials science. *Chinese Physics B*, 32(11):118104.

Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. 2024. xrag: Extreme context compression for retrieval-augmented generation with one token. *Advances in Neural Information Processing Systems*, 37:109487–109516.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Tri Dao. 2024. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and 1 others. 2023a. Rarr: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023b. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, and 1 others. 2023. Language is not all you need: Aligning perception with language models. *Advances in Neural Information Processing Systems*, 36:72096–72109.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: main volume*, pages 874–880.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Kelvin Jiang, Dekun Wu, and Hui Jiang. 2019. FreebaseQA: A new factoid QA data set matching trivia-style question-answer pairs with Freebase. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 318–323, Minneapolis, Minnesota. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. triviaqa: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. *arXiv e-prints*, arXiv:1705.03551.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, and 1 others. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*.

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

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Haitao Li, Qingyao Ai, Jia Chen, Qian Dong, Zhijing Wu, and Yiqun Liu. 2025. Blade: Enhancing black-box large language models with small domain-specific models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24422–24430.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947.

Vladislav Lialin, Vijeta Deshpande, and Anna Rumshisky. 2023. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

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

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024b. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi, and Daniel Khashabi. 2022. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. *arXiv preprint*.

Yuning Mao, Lambert Mathias, Rui Hou, Amjad Almahairi, Hao Ma, Jiawei Han, Scott Yih, and Madian Khabsa. 2022. Unipelt: A unified framework for parameter-efficient language model tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6253–6264.

Junbo Niu, Zheng Liu, Zhuangcheng Gu, Bin Wang, Linke Ouyang, Zhiyuan Zhao, Tao Chu, Tianyao He, Fan Wu, Qintong Zhang, Zhenjiang Jin, Guang Liang, Rui Zhang, Wenzheng Zhang, Yuan Qu, Zhifei Ren, Yuefeng Sun, Yuanhong Zheng, Dongsheng Ma, and 42 others. 2025. Mineru2.5: A decoupled vision-language model for efficient high-resolution document parsing. *Preprint*, arXiv:2509.22186.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. Adapterfusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th conference of the European chapter of the association for computational linguistics: main volume*, pages 487–503.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. Replug: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8371–8384.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1–9.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adalora: Adaptive budget allocation for parameter-efficient finetuning. *arXiv preprint arXiv:2303.10512*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Lucia Zheng, Neel Guha, Javokhir Arifov, Sarah Zhang, Michal Skreta, Christopher D Manning, Peter Henderson, and Daniel E Ho. 2025. A reasoning-focused legal retrieval benchmark. In *Proceedings of the 2025 Symposium on Computer Science and Law*, pages 169–193.

A Dataset Details

Table 5 summarizes the datasets used in our study. For general-domain QA, we report EM on six public benchmarks—FreebaseQA (Jiang et al., 2019), HotpotQA (Yang et al., 2018), Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), WebQuestions (Berant et al., 2013), and PopQA (Mallen et al., 2022). Because the official dev/test splits of these benchmarks are substantially larger than needed to characterize general QA behavior under a fixed inference setup, we evaluate on a single fixed random subset from each dataset's official dev/test split and reuse the same subsets across all methods, ensuring a consistent, reproducible, and benchmark-balanced probe of general capability. For specialist knowledge injection, we use two expert benchmarks in immunology adjuvant and catalytic materials (Anonymous, 2025b,a), paired with private literature corpora (813 papers for adjuvants; 986 papers for materials). We evaluate specialist QA with BERTScore (Zhang et al., 2019) computed using SciBERT (Beltagy et al., 2019) to better reflect semantic faithfulness in scientific language.

B Additional Implementation Details

All experiments are run on 8×NVIDIA A100 GPUs with bfloat16 precision (FlashAttention-2 is enabled when available). We report the complete hyperparameter settings for the two training stages in Table 6 (Stage I) and Table 7 (Stage II) to facilitate reproducibility across domains.

For PPR, query representations are obtained by attention-masked mean pooling over the encoder's last-layer states followed by ℓ_2 normalization. Prototype banks are built using 32 prototypes per domain, subsampling up to 10k in-domain queries for clustering, and setting n_init=10.

Unless otherwise stated, we disable the model's explicit "thinking" mode during both training and evaluation (i.e., we use the non-thinking chat format). At inference, we apply nucleus/top-k sampling for both background synthesis and answer generation with temperature 0.7, top-p 0.8, and top-k 20; the maximum generation lengths follow the benchmark configuration.

C Prompt Templates

To ensure a controlled and reproducible evaluation, we employ a small set of fixed chat-style prompt templates across all experiments. The templates are intentionally minimal to reduce incidental prompt variance, while still enforcing domain-appropriate behavior for expert routes. Unless otherwise stated, we disable the model's explicit "thinking" mode and use the non-thinking chat format throughout.

General route. Figure 6 shows the prompt used for the general route, where the frozen base model LLM_{base} answers directly without any domain injection. This template is deliberately concise to serve as a clean probe of the base model's general QA capability under a stable instruction format.

Domain routes. For each private domain i, prompting is factorized into two stages: (i) a background-synthesis prompt for the domain expert $\text{LLM}_{\text{domain},i}$ to elicit domain-relevant context conditioned on the user question, and (ii) an answering prompt for the frozen base model LLM_{base} that produces the final response. Figures 7 and 8 present the background-synthesis prompts for the adjuvant and materials experts, respectively. Figures 9 and 10 present the corresponding answering prompts for LLM_{base}.

```
[User]
Please answer the following question.
Question: <query>
```

Figure 6: **General-route prompt template.** The frozen base model LLM_{base} answers the query directly using a minimal instruction format. Placeholders (shown in blue) indicate runtime fields (e.g., `<query>`).

[System]
You are an expert in immunology and adjuvant, with a strong background in vaccine development. Your research and practice in this field have equipped you with a deep understanding of the mechanisms of immune response and how to optimize vaccine efficacy through adjuvants. You excel in providing relevant and professional background knowledge that can help answer the question. Please provide the background knowledge related to the following question.

[User]
Question: <question>

Figure 7: **Adjuvant-domain background-synthesis prompt.** Template for the domain expert $\mathrm{LLM}_{\mathrm{domain,adj}}$ to generate domain-relevant background knowledge conditioned on the user question. Placeholders (shown in blue) are filled at runtime.

[System]
You are an expert in immunology and adjuvant, with a strong background in vaccine development. Your research and practice in this field have equipped you with a deep understanding of the mechanisms of immune response and how to optimize vaccine efficacy through adjuvants. You excel in providing concise, precise, and professional responses to questions related to adjuvants and immunology.

[User]
Please answer the following question based on the knowledge provided.
Question: <query>
Knowledge: <background knowledge>

Figure 9: **Adjuvant-domain answering prompt with GAG injection.** Template for the frozen base model $\mathrm{LLM}_{\mathrm{base}}$ under the adjuvant expert route. The `Knowledge:` field denotes the single-token injection slot in GAG: it is instantiated by a one-token, retrieval-free injected interface. This contrasts with RAG-style prompting that serializes chunked top-k evidence into text and can introduce fragmented or incomplete context. Placeholders (shown in blue) are filled at runtime.

[System]
You are an expert in materials science and engineering. Your research and practice have equipped you with a deep understanding of how composition, structure, processing and environment determine material properties and performance. You excel in providing relevant and professional background knowledge that can help answer research-level questions in this field. Please provide the background knowledge related to the following question. Do not fabricate specific numerical data that is not generally known in the field.

[User]
Question: <question>

Figure 8: **Materials-domain background-synthesis prompt.** Template for the domain expert $\mathrm{LLM}_{\mathrm{domain,mat}}$. Placeholders (shown in blue) are filled at runtime.

[System]
You are an expert in materials science and engineering, with extensive experience in the design, synthesis, and characterization of functional materials. Your research and practice have equipped you with a deep understanding of structure–property relationships, reaction mechanisms, and performance optimization strategies. You excel in providing concise, precise, and professional responses to questions related to materials design, processing, and applications.

[User]
Please answer the following question based on the knowledge provided.
Question: <query>
Knowledge: <background knowledge>

Figure 10: **Materials-domain answering prompt with GAG injection.** Template for the frozen base model $\mathrm{LLM}_{\mathrm{base}}$ under the materials expert route. Placeholders (shown in blue) are filled at runtime.

Category	Dataset	Split / Size	Metric
General QA	FreebaseQA (Jiang et al., 2019)	Eval: 1135	EM
	HotpotQA (Yang et al., 2018)	Eval: 1135	EM
	Natural Questions (Kwiatkowski et al., 2019)	Eval: 1135	EM
	TriviaQA (Joshi et al., 2017)	Eval: 1135	EM
	WebQuestions (Berant et al., 2013)	Eval: 1135	EM
	PopQA (Mallen et al., 2022)	Eval: 1135	EM
Private expert QA	Adjuvant (Anonymous, 2025b)	Train: 38,277 Test: 1,135 Corpus: 813 papers	BERTScore (SciBERT)
	Materials (Anonymous, 2025a)	Train: 3,661 Test: 646 Corpus: 986 papers	BERTScore (SciBERT)

Table 5: Dataset summary. "Eval" denotes the number of the evaluated questions used for general QA benchmarks. Expert benchmarks are paired with private literature corpora and evaluated with BERTScore using SciBERT.

Hyperparameter	Value
Optimizer	AdamW (β_1=0.9, β_2=0.999, ϵ=10^{-8})
Learning rate	5×10^{-6}
LR schedule	cosine, warmup ratio 0.1
Epochs	8
Per-device train batch size	8
Gradient accumulation	2
GPUs	8
Global train batch size	128
Precision	bfloat16
Seed	42

Table 6: Stage I hyperparameters for domain expert adaptation.

Hyperparameter	Value
Max sequence length	2048
Optimizer	AdamW
Learning rate	6×10^{-3}
LR schedule	linear, warmup ratio 0.03
Weight decay	0.0
Epochs	5
Per-device train batch size	1
Gradient accumulation	4
GPUs	8
Global train batch size	32
Precision / acceleration	bfloat16; FlashAttention-2 when available
Seed	980406

Table 7: Stage II hyperparameters for projector alignment.

D RAG Sensitivity to Retrieval Depth

As shown in Table 8, we further examine how retrieval depth affects RAG in specialized private domains by varying k, the number of retrieved passages concatenated to the base-model prompt. For brevity, Table 1 reports the best-performing k, and we provide the full sweep here.

Discussion. Increasing k does not monotonically improve RAG on either domain; in our setting, k=1 is best and larger k slightly degrades performance. This behavior is consistent with (i) long-context interference, where additional retrieved text increases distraction and dilutes the signal needed for generation, and (ii) retrieval brittleness in specialized corpora, where adding more passages may amplify partially mismatched or fragmented evidence rather than improving reference-critical coverage. By contrast, GAG avoids prompt-time evidence serialization and provides a constant-budget injection interface, yielding substantially stronger specialist QA with more predictable inference behavior.

System	Top-k	BERTScore↑ (SciBERT)	
		Adjuvant	Materials
Base-Model-Only (Qwen3-8B)	0	56.12	60.01
RAG (Qwen3-8B)	1	59.97	62.13
	3	58.68	61.17
	5	58.76	60.87
	7	58.27	60.10
	9	58.22	60.29
GAG (Gen+Adj+Mat)	–	**69.17**	**71.36**

Table 8: **Effect of retrieval depth in RAG.** We report BERTScore (SciBERT) on Adjuvant and Materials when concatenating the top-k retrieved passages.

E Additional Results: Incremental Multi-domain Routing with PPR

Section 6.2 shows that PPR enables reliable selective activation in the main routed setting. Here we further examine incremental multi-domain expansion under a strictly plug-and-play protocol. Starting from *General+Adjuvant*, we progressively add *Materials*, *Aviation* (Agarwal et al., 2022), *Law* (Zheng et al., 2025), and *Math* (Cobbe et al., 2021). Crucially, at each step we construct and load only the prototype bank of the newly introduced route, while keeping the query encoder and all previously deployed prototype banks fully frozen. Routing is performed by nearest-prototype cosine similarity, so performance directly reflects how well domain query manifolds remain separable in a shared embedding space without router retraining or threshold tuning.

Route	Source / composition	#Queries
General	FreebaseQA (189) HotpotQA (196) Natural Questions (193) TriviaQA (184) WebQuestions (183) PopQA (190)	1,135
Adjuvant	Adjuvant expert QA	1,135
Materials	Materials expert QA	646
Aviation	sakharamg/AviationQA	1,135
Law	reglabs/housing_qa	1,135
Math	openai/gsm8k	1,135
Total	—	**6,321**

Table 9: **Evaluation pool for incremental multi-domain routing.** The General route is a balanced union of six public QA subsets (sizes in parentheses).

Evaluation composition. Table 9 summarizes the evaluation pool (6,321 queries in total) spanning six routes. The General route is constructed as a balanced union of six public QA subsets (1,135 queries), while each specialist route contributes a domain-specific evaluation set of comparable scale (except Materials, 646 queries).

#Routes	New domain	Micro acc. (%)	Per-route acc. (%)					
			Gen	Adj	Mat	Avi	Law	Math
2	—	99.78	99.65	99.91	—	—	—	—
3	+ Mat	99.55	99.65	99.38	99.69	—	—	—
4	+ Avi	99.63	99.47	99.38	99.69	100.00	—	—
5	+ Law	99.71	99.47	99.38	99.69	100.00	100.00	—
6	+ Math	99.72	99.47	99.38	99.69	100.00	100.00	99.74

Table 10: **Incremental multi-domain scalability of PPR.** We progressively add new routes by loading only the corresponding prototype bank (encoder and existing routes are frozen). Micro acc. is overall routing accuracy; Per-route acc. is class-wise accuracy for each active route.

Incremental routing stability. Table 10 reports routing accuracy as the active route set grows from 2 to 6. PPR exhibits strong scalability under incremental expansion: micro-averaged accuracy remains above 99.5% across all stages, despite repeatedly increasing the decision space. Moreover, per-route accuracy stays uniformly high for both newly added routes and previously deployed routes, with no meaningful degradation as new prototype banks are attached. Together, these results position PPR as a non-parametric, deployment-friendly routing interface for modular expert systems: domain expansion is realized by a lightweight prototype update, preserving the stability guarantees of a frozen base model while maintaining near-saturated routing reliability at scale.

F More Interesting Cases

Note on visualization. In Figures 11–14, the "Generated Expert Background" on the GAG side is shown only as an analysis-time probe for interpretability. In the actual GAG pipeline, the domain expert model ($LLM_{domain,i}$) does **not** expose any such text to users; it produces a **single continuous embedding** that is projected into LLM_{base}'s token space and injected as **one token** (cf. §4.2). Thus, the user-visible interface remains constant-budget and retrieval-free.

Legend (highlight colors). Across Figures 11–14, green highlights denote ground-truth-critical key factors, red highlights mark off-target/mismatched or misleading retrieved details that can derail RAG, and gray text indicates irrelevant/noisy content that is not required by the reference answer.

Case 1 (Adjuvant): noisy retrieval → incomplete evidence grounding. As shown in Figure 11, this case illustrates a practical brittleness of RAG in private scientific corpora: the retrieved top passage is partially corrupted and fragmented, so the base model is forced to answer under incomplete and unstable evidence support. Even when the RAG answer captures the coarse direction (Th1 bias), retrieval noise can suppress explicit coverage of all reference-critical markers. GAG avoids this failure mode by decoupling domain knowledge transfer from snippet quality: a single injected expert token provides a holistic domain prior in LLM_{base}'s representational space, enabling more reliable coverage of the key Th1 evidence emphasized by the ground truth.

Case 2 (Adjuvant): retrieval drift → objective misalignment. As shown in Figure 12, this example exposes a high-stakes RAG failure mode in private scientific corpora: objective-misaligned evidence can be topically relevant yet steer generation toward the wrong criterion. The ground truth is explicitly titer-based (higher env-specific IgG/IgA under DNA–VLP than VLP–VLP) and attributes the gap to a coherent prime–boost mechanism. However, retrieved snippets foreground adjuvant/mucosal and neutralization-related details (plus off-target readouts like anti-gag), so the base model over-focuses on epitope breadth/neutralization narratives and under-serves the titer-focused comparison the question demands. GAG avoids this drift by replacing snippet-level evidence serialization with a representation-level expert prior: a single injected token encodes the causal chain needed for the titer claim, delivering higher intent fidelity under a constant knowledge budget.

Case 3 (Materials): wrong-entity retrieval → mechanism collapse. As shown in Figure 13, mechanism questions are particularly vulnerable to RAG's entity mismatch: when retrieval locks onto an adjacent but different experimental setup (here, Cu/Au catalyst characterization), the base model is steered into an evidence-consistent yet question-inconsistent explanation. Consequently, RAG shifts to a Cu/Au-specific story and fails to cover the ground-truth mechanism checklist (transport effects, overlapping-field charge transfer, intermediate stabilization, and aggregation/sintering tradeoffs). GAG mitigates this by injecting a domain-conditioned representation that is not tied to a single retrieved entity or paper chunk, allowing LLM_{base} to synthesize the intended cross-concept mechanism and maintain high-level faithfulness under retrieval mismatch.

Case 4 (Adjuvant; error analysis): a minor numeric-scale slip. As shown in Figure 14, this error case reflects a common pattern in scientific QA: retrieval can directly surface and copy exact numeric details when they appear verbatim in the retrieved span. GAG still provides strong procedural correctness (the preparation method aligns at a high level), but shows a small unit/scale slip (nm vs. μm) on the mean size. In practice, this is a lightweight edge case: when exact numeric fidelity is paramount, simple post-hoc numeric/unit normalization can be layered on top of the injected expert signal without changing the core constant-budget design.

Question: How does the Fullerenol-Env complex compare to naked Env in terms of inducing Th1-biased immune responses?

Ground Truth: The **Fullerenol-Env complex** induces a **stronger Th1-biased immune response** compared to **naked Env**, as evidenced by elevated levels of Th1 cytokines like **IFN-γ and TNF-α**.

RAG Approach

Retrieved Evidence

imag of Env erppd by fl and th ant propet of fuee D-1). Rd al r and whit for on th fullerene surface and green ball rpresent C atoms. () **Compared to naked Env immunization group, IFN- production (immunospot) was signifcantly hanced** when mice immunized with the fomulation via vrios imunizatio routes, incuding intrademal (i.), inrauscular (i.), suuta neous (s.) and inranasal (in.) inctions. Fleenl could decrae the antigen dage () and mmunization time ().) FN-odction in mice immunized thre imes with naked E DNA or twice with the FullerenlEnv complex via inrademal injection while h later inducing een T cells proliferation capability at different time points post third immunization with Fullerenol-Env complex (20 μg - 20 μg / mouse). (1) Fullerenol-Env analyzed by Student's t-test.
We investigated the adjuvant activities of fullerenol in mice by using DNA encoding the HIV-1 envelope protein gp145 (Env) as the antigen, which has already entered Phase I clinic trials in China after receiving satisfied safety assessment from the good laboratory practice toxicology study in animal models (data not shown). We found that the formulation of FullerenolEnv complex enhanced the cellular immunity when administered via various immunization routes. Using intradermal administration, less DNA antigen and fewer immunizations were required to achieve a similar level of cellular immunity to that induced by DNA alone. The formulation could also promote the innate immunity. Meanwhile, the effector memory T cells (T_EM) -biased immune response was induced by this formulation. The formulation induced these responses by triggering polyvalent immunities via the activation of multiple Toll-like receptor (TLR) signaling pathways. This is the first report showing that the nanoparticle of fullerenol designed as a viruslike particle can be used as an effective vaccine nano-adjuvant.

Final Answer

The Fullerenol-Env complex induces a **Th1-biased** immune response more effectively than naked Env. This is evidenced by the significantly **enhanced production of IFN-γ** (a hallmark of Th1 responses) in mice immunized with the Fullerenol-Env complex compared to those immunized with naked Env. The adjuvant effect of Fullerenol is attributed to its ability to:Enhance antigen presentation by reducing antigen degradation and improving immunization efficiency.Trigger polyvalent immune activation through the activation of multiple Toll-like receptor (TLR) signaling pathways.Induce effector memory T cells (T_EM), which are associated with strong and long-lasting cellular immunity.Require fewer doses and lower antigen amounts to achieve comparable or superior immune responses, particularly when administered via intradermal routes.In summary, the Fullerenol-Env complex outperforms naked Env in promoting Th1-biased immune responses, making it a promising candidate as a vaccine adjuvant. ✗

GAG Approach

Generated Expert Background

The Fullerenol-Env complex induces a **Th1-biased** immune response, characterized by **higher IFN-γ** production and lower IL-4 production compared to naked Env, which is crucial for effective HIV-1 control.

Final Answer

The Fullerenol-Env complex induces a **more Th1-biased immune response compared to naked Env**, as evidenced by **higher levels of IFN-γ and TNF-α production in splenocytes.** ✓

Figure 11: **Case 1 (Adjuvant): Robust key-factor coverage under noisy/fragmented retrieval.** RAG retrieves a corrupted/noisy snippet, which weakens evidence completeness for reference-critical cytokine signals, while GAG yields a more faithful Th1 comparison via constant-budget expert-token injection. Green highlights indicate reference-critical factors.

Question: How does the **heterologous DNA-VLP prime-boost protocol** compare to the **homologous VLP-VLP protocol** in terms of inducing **env-specific antibody responses**, and what **potential mechanisms** could explain any differences observed?

Ground Truth: The heterologous DNA-VLP prime-boost protocol induced **significantly higher env-specific serum IgG and IgA titers** compared to the homologous VLP-VLP protocol. This enhanced response likely stems from several factors: 1) **DNA vaccination efficiently primes T helper cells**, providing robust help for B cell responses upon protein boost. 2) The DNA prime may **activate a broader repertoire of naive B cells** due to intracellular expression and processing of the env protein. 3) The **VLP boost provides a potent stimulus for memory B cells** induced by DNA priming. 4) **Using env antigens from different clades (B for DNA, A for VLP)** may broaden the epitope specificity of the antibody response. 5) The DNA prime followed by VLP boost **mimics natural infection** (initial antigen expression followed by whole virus exposure), potentially eliciting more functionally relevant antibodies. This synergistic effect highlights the potential of heterologous prime-boost strategies to enhance HIV vaccine immunogenicity.

RAG Approach

Retrieved Evidence

The designed heterologous protocol is able to increase the env-specific humoral and cellular immune response, compared to the homologous protocol, which is to some extent increased by the administration of **L3-adjuvanted VLP boosting dose.** The **anti-gag response** is statistically increased in both homologous and heterologous protocols, particularly when the VLP boosting dose is adjuvanted. Immune sera from immunized animals exhibit **50 % ex vivo neutralizing activity** against **heterologous A and B-clade viral isolates.** An **envelope B-cell epitope mapping** shows an enhanced response against **V3 epitopes** all across the **C2–V5 region** in the heterologous prime–boost immunization strategy.**The induction of humoral immunity at mucosal sites, which represents the main port of entry for the HIV-1 infection, is extremely relevant. In this framework, the DNA–VLP heterologous prime–boost protocol appears a promising preventive vaccine approach which can significantly benefit from specific mucosal adjuvants, as the Eurocine L3.**

Final Answer

The heterologous DNA-VLP prime-boost protocol demonstrates a superior ability to induce env-specific antibody responses compared to the homologous VLP-VLP protocol, **particularly in terms of neutralizing activity and B-cell epitope recognition.** Here's a breakdown of the comparison and potential mechanisms: 1. Env-Specific Antibody Responses - Heterologous DNA-VLP: - Induces enhanced humoral immunity, including **stronger neutralizing activity** against heterologous A and B-clade viral isolates. - Shows **enhanced response against V3 epitopes across the C2–V5 region** of the envelope protein. - Benefits from **mucosal adjuvants like Eurocine L3**, which may **enhance mucosal immune responses**—critical for HIV-1 prevention, as mucosal sites are the primary entry points. - Homologous VLP-VLP: - While it also induces env-specific antibodies, the neutralizing activity and epitope breadth are less robust compared to the heterologous protocol. - **The L3-adjuvanted boosting dose** can partially enhance the response, but not to the same extent as the heterologous approach. 2. Potential Mechanisms for Enhanced Responses in Heterologous Prime-Boost - Immune Diversification: - The DNA prime may present different antigenic epitopes compared to the VLP boost, leading to **diversified B-cell activation and enhanced epitope coverage.** - This can result in a more **broad and cross-clade neutralizing response.** - Adjuvant Effects: - **The mucosal adjuvant L3** enhances mucosal immunity, which is crucial for HIV-1 prevention. - It may also enhance the quality of the antibody response, including **neutralizing potency and epitope recognition.** - Immune Memory and Potency: - The prime-boost strategy can **amplify immune memory** by activating different immune cell subsets (e.g., T cells and B cells) at different stages. - The DNA prime may induce stronger antigen-presenting cell (APC) activation, leading to **more effective T-cell help for B-cell responses** during the VLP boost. - **Reduced Antigenic Escape:** - The heterologous approach may reduce the risk of immune escape by targeting conserved epitopes (e.g., V3) across different HIV-1 clades. 3. Conclusion The heterologous DNA-VLP prime-boost protocol outperforms the homologous VLP-VLP protocol in inducing env-specific antibody responses, **particularly in terms of neutralizing activity and epitope breadth.** This is likely due to immune diversification, enhanced adjuvant effects, and **improved mucosal immunity.** These findings support the promising potential of the DNA-VLP heterologous strategy as a preventive HIV-1 vaccine approach.

- **Off-target evidence → answer drifts**
- **Misses IgG/IgA titer focus**

GAG Approach

Generated Expert Background

The heterologous DNA-VLP prime-boost protocol induced higher levels of env-specific antibodies compared to the homologous VLP-VLP protocol. In many settings, the heterologous sequence is associated with **stronger serum env-specific responses (including IgG and sometimes IgA),** while repeated VLP-only immunization may rely more on formulation/adjuvanting to reach similar magnitude. Some reports also suggest a modest shift in antibody quality/subclass balance with DNA priming (often discussed as a more "cell-help–supported" profile), although the exact pattern can vary by model and antigen. These differences in antibody profiles could be explained by several potential mechanisms:1)More effective **priming of CD4 T-cell help by DNA vaccination,** improving the magnitude and quality of B-cell responses upon boosting.2)**In vivo antigen expression from the DNA prime may expose env through endogenous processing and presentation,** potentially recruiting a **broader initial B-cell repertoire.**3)The VLP boost provides a **multivalent particulate display of env** that can efficiently **expand primed/memory B cells and support affinity maturation.**4)If prime/boost involve **heterologous env variants (e.g., different clades/constructs), responses may broaden across epitopes** rather than remaining narrowly strain-matched.5)The **prime–boost order** may favor **functionally relevant antibody specificities** (often discussed in terms of neutralization breadth and common regions like V3/C2–V5), though this can be assay- and isolate-dependent.6)Finally, homologous VLP–VLP can be partially improved by stronger adjuvanting/optimized dosing, but may still lag if early priming signals are weaker or less diverse.

Final Answer

The heterologous DNA-VLP protocol induced **higher env-specific antibody responses** compared to the homologous VLP-VLP regimen. This difference can be attributed to several potential mechanisms: 1) The DNA prime can provide an **effective priming phase with sustained in vivo antigen expression,** helping establish **stronger T-cell help** for subsequent B-cell expansion. 2) DNA priming may **recruit a broader set of naive B cells** through **endogenous antigen processing/presentation,** setting up a more diverse starting repertoire. 3) The VLP boost then delivers a **strong multivalent particulate stimulus** that efficiently **expands primed/memory B cells** and **supports affinity maturation.** 4) When the prime and boost are not identical in env design **(e.g., heterologous variants/clades), this can broaden epitope coverage** rather than simply amplifying one matched specificity. 5) Overall, **the heterologous sequence may bias responses toward more functional antibody features (e.g., neutralization-associated epitopes or mucosal-relevant IgA in some settings)**, while VLP–VLP often benefits more from adjuvant/formulation to partially close the gap.

- ✓ **Mechanism-aligned, titer-focused**
- ✓ **Retrieval-free robustness**

Figure 12: **Case 2 (Adjuvant): Titer-critical QA under retrieval drift.** The question targets env-specific serum IgG/IgA titers and their mechanistic drivers. RAG retrieval over-emphasizes Eurocine L3/mucosal framing, anti-gag signals, and neutralization/epitope-mapping anecdotes, inducing a neutralization-centric answer that drifts from the requested titer comparison. GAG instead yields a titer-aligned, mechanism-grounded explanation (DNA priming for CD4 help and broadened naive B-cell recruitment; VLP boosting for strong memory B-cell expansion/affinity maturation; heterologous clades broadening epitope coverage) via expert-token injection.

Question: In the context of utilizing an ensemble of Cu nanoparticles for CO_2 reduction, how do intermolecular interactions within the nanoparticle ensemble affect reaction kinetics and overall efficiency?

Ground Truth: Intermolecular interactions within an ensemble of Cu nanoparticles can facilitate or hinder reaction kinetics by affecting **mass transport properties** and the **stabilization of key reaction intermediates**. The proximity of **overlapping electric fields** may **facilitate charge transfer** and **lower the energy barrier** for electron and proton transfer to aqueous CO2. Additionally, kinetic models should consider the role of **nanoparticle aggregation**; **well-dispersed NPs maximize active surface area** while **excessive sintering can lead to reduced catalytic activity.**

RAG Approach

Retrieved Evidence

Quasi-in-situ XPS 41-43 was performed to **study the surface structure of the Cu/Au catalyst.** The surface **Au sites are in the Au^0 state**, as evidenced by the binding energy of the Au 4f electrons (Figure 4a). The binding energy of the Cu 2p electrons and the absence of their satellite peaks demonstrate that the Cu2+ from the original CuO nanoparticles has been fully reduced (Figure 4b). **Given that Cu1+** and Cu^0 cannot be distinguished in the Cu 2p XPS spectrum (Figure S8), we acquired the Cu LMM Auger spectrum to analyze the oxidation state. While the surface of the Cu catalyst is dominated by Cu^0, the Cu/Au catalyst contains a significant number of **Cu1+ sites** (Figure 4c). Further experiments show that the oxidation states of the Au and Cu sites are retained after long-term electrolysis (Figure S9). These results suggest that the **Cu-Au metal-metal interactions** stabilize Cu1+ on the surface under reducing potentials, which is likely related to the enhanced performance for CO2 reduction 18,44. Figure 4. (a) Au 4f XPS spectra of the Cu/Au catalyst and Au nanoparticles. (b) Cu 2p XPS spectra and (c) Cu LMM auger spectra of the Cu/Au and Cu catalysts. In summary, we have developed a **formate-selective** CO2 reduction electrocatalyst based on an intimate mixture of Cu and Au nanoparticles. The metal-metal interactions make the Cu **15 times more active and 4 times more selective toward CO2 conversion to formate**; the **Au is deactivated for converting CO2 to CO.** These findings highlight the promise of utilizing inter-nanoparticle metal-metal interactions to improve CO2 reduction electrocatalysis.

Final Answer

In the context of utilizing an ensemble of Cu nanoparticles for CO_2 reduction, intermolecular interactions—specifically **metal-metal interactions between Cu and Au nanoparticles**—play a critical role in modulating reaction kinetics and overall efficiency. These interactions influence the electronic structure and surface chemistry of the catalyst, which in turn affect the catalytic performance. 1. Stabilization of Cu^{1+} Sites: The **Cu/Au ensemble stabilizes Cu^{1+} sites on the surface under reducing potentials,** as evidenced by XPS and Auger spectroscopy. **This is crucial because Cu^{1+} is known to be more active for CO_2 reduction compared to Cu^0,** and its presence enhances the **selectivity toward formate production.** 2. Enhanced Activity and Selectivity: **The metal-metal interactions between Cu and Au lead to a significant increase in the activity of Cu (15 times)** and **selectivity toward formate (4 times) compared to bare Cu nanoparticles.** This is attributed to the synergistic effects of Au's electronic properties and Cu's catalytic behavior, which optimize the reaction pathway for CO_2 reduction. 3. Retention of Oxidation States: The oxidation states of both Cu and Au are retained during long-term electrolysis, indicating the stability of the Cu/Au ensemble. This stability ensures consistent performance over time, which is essential for practical electrocatalytic applications. 4. **Deactivation of Au for CO Production: While Au is typically active for CO production, in the Cu/Au ensemble, Au is deactivated for this pathway. This selective deactivation enhances the overall efficiency of the catalyst by directing the reaction toward the more desirable formate product.** In summary, intermolecular interactions in the Cu/Au nanoparticle ensemble significantly enhance the reaction kinetics and overall efficiency of CO_2 reduction by stabilizing active Cu^{1+} sites, improving selectivity, and ensuring long-term stability. These findings highlight the potential of metal-metal interactions in designing advanced electrocatalysts for CO_2 conversion.

Missing (vs. GT):
- **mass transport**
- **overlapping electric fields**
- **aggregation / sintering**

GAG Approach

Generated Expert Background

Intermolecular interactions within the nanoparticle ensemble can significantly influence reaction kinetics by affecting **mass transport and local concentration gradients of reactants.** As nanoparticles are often in close proximity, they can create **localized electric fields** that **enhance the adsorption of CO_2 and other intermediates**, facilitating the reaction pathway. Moreover, these interactions can lead to cooperative effects where the presence of one nanoparticle influences the activity of another, potentially leading to optimized catalytic behavior through **enhanced charge transfer** and **reduced energy barriers** for **intermediate stabilization.** This cooperative effect can improve the overall efficiency of the ensemble, demonstrating how the **spatial arrangement and proximity** of catalytic sites can enhance performance.

Final Answer

Intermolecular interactions within an ensemble of Cu nanoparticles can facilitate or hinder reaction kinetics by affecting **mass transport properties and the stabilization of key reaction intermediates.** The proximity of NPs can create a microelectrochemical environment where **charge transfer is accelerated** due to **overlapping electric fields**, thereby **lowering the activation energy** for electron transfer to aqueous CO_2. Additionally, **kinetic models should consider the role of nanoparticle aggregation**; **well-dispersed NPs maximize active surface area** while **excessive aggregation can lead to increased mass transport resistance.**

Covers GT:
✓ **transport**
✓ **fields**
✓ **intermediates**
✓ **aggregation**

Figure 13: **Case 3 (Materials): Mechanism-level synthesis under wrong-entity retrieval.** RAG retrieval centers on a Cu/Au-XPS characterization narrative, inducing an off-target answer that misses the reference mechanisms (mass transport, overlapping fields, intermediate stabilization, aggregation effects). GAG recovers the intended ensemble-interaction mechanisms via one-token injection and explicitly covers the reference factors.

Question: What method was used to prepare PLA microparticles, and what was their mean size?

Ground Truth: PLA microparticles were prepared using the oil-in-water emulsion–solvent evaporation method combined with the SPG membrane emulsification technique, and their mean size was **800 nm.**

RAG Approach

Retrieved Evidence

PLA (poly(lactic acid)) microparticles (MPs) were prepared by the oil-in-water emulsion–solvent evaporation method combined with the SPG membrane emulsification technique. PLA MPs exhibited spherical shapes, with a mean size of 800 nm. The surface charge of MPs was negative (−17.05 mV). As shown in Fig. 1A, MPs with different positive surface charges were prepared by coating PLA MPs with different cationic polymers (CS, chitosan, 8.67 mV; CSC, chitosan hydrochloride, 20.12 mV; PEI, polyethylenimine, 45.27 mV). Vaccine formulations were prepared by incubating MPs with antigen solution overnight. With the MP surface charge increasing, the adsorption efficiency of antigen onto MPs increased. In vitro cell experiments suggested that enhancing positive surface charge of MPs favored antigen uptake by antigen-presenting cells (Raw 264.7 macrophages and bone-marrow derived dendritic cells). Animal immunization study indicated that both humoral immune response (Fig. 1B) and cellular immune response (Fig. 1C) could be improved by regulating the surface charges of MPs. Understanding how particle surface charges affect the efficacy of MP-based vaccine might have significant implications for rational vaccine design.

Final Answer

The PLA microparticles were prepared using the oil-in-water emulsion–solvent evaporation method combined with the SPG membrane emulsification technique. The mean size of the PLA microparticles was **800 nm.**

GAG Approach

Generated Expert Background

PLA microparticles were prepared using the emulsion solvent evaporation method. The mean particle size was 1.15 ± 0.08 µm, which is in the range of 1-2 µm, suitable for phagocytosis by antigen-presenting cells.

Final Answer

PLA microparticles were prepared by emulsification and solvent evaporation, resulting in a mean size of **5.6 µm.**

Figure 14: **Case 4 (Adjuvant; error analysis): Exact numeric fidelity vs. knowledge transfer.** RAG can precisely copy an explicitly stated mean size from retrieved text, while GAG correctly captures the preparation procedure but exhibits a minor nm/μm scale slip on the numeric value. This suggests that lightweight numeric normalization/verification can complement constant-budget injection when exact numbers dominate the evaluation.