DIAGNOSE, ALLOCATE, AND BIND: CONDITIONAL KNOWLEDGE INJECTION INTO LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Although language models can assimilate target knowledge through continued pre-training, conditional knowledge injection remains fundamentally limited by uniform token exposure and condition-independent recall shortcuts, where models often recall family-consistent values without binding them to their governing conditions. Under a fixed training-token budget, this exposes two coupled challenges: identifying facts requiring additional exposure and ensuring recalled values depend strictly on their queried conditions. This motivates a natural question: *Can fact-level internal states be diagnosed to dynamically allocate exposure while explicitly binding values to their applicability conditions?* To this end, we propose **DAB** (**D**iagnose, **A**llocate, and **B**ind), a framework linking fact-level memory diagnosis to adaptive token allocation and condition-dependent supervision. DAB exactly decomposes the gold-value negative log-likelihood into family-mass and within-family concentration terms, isolating under-supported facts from those confused with sibling values via behavior-calibrated routing. After a shared uniform stage, DAB reallocates a bounded token budget toward facts with diagnosed learning needs while strictly preserving minimum exposure for every fact. Furthermore, a late-stage Condition Necessity objective enforces a reference-relative margin between full-condition and condition-masked inputs, explicitly penalizing condition-independent shortcuts. Experiments on domain-specific technical standards (GB-Standard) and SQuAD-derived closed-book QA across Qwen3.5-4B-Base and Llama-3.1-8B-Instruct demonstrate the generality of DAB. On GB, DAB improves closed-book question-level accuracy by up to **8.23 percentage points** over the same-renderer uniform baseline. On SQuAD, DAB reaches **64.06%** with Qwen and **69.64%** with Llama, outperforming the strongest baseline, Knowledge-Instruct (KI), by **6.22** and **19.50 percentage points**, respectively. Factorial ablations confirm that diagnosis-guided allocation and conditional supervision provide complementary gains, framing conditional knowledge injection as the joint problem of allocating exposure and learning condition–value bindings.

1 INTRODUCTION

Large Language Models (LLMs) have demonstrated impressive capabilities in assimilating factual knowledge through continued pre-training and supervised fine-tuning (Gururangan et al., 2020). However, in domain-specific technical standards, legal statutes, and situated environments, factual correctness is rarely unconditional: a single query key often admits different valid values depending on operating environments, configurations, or time periods (Zhang & Choi, 2021; Dhingra et al., 2022). In these conditional settings, factual recall is still predominantly treated as uniform token prediction over raw text. As a result, models may successfully recall plausible values belonging to a factual category yet fail to select the specific value licensed by the queried condition. In this sense, current models conflate *acquisition failure* (failing to support any candidate value) with *binding failure* (failing to discriminate among competing sibling values), leading to brittle memorization and condition-independent shortcuts.

To address the challenge of factual knowledge injection, recent studies have explored two major paradigms: *Unconditional Corpus Augmentation* and *External Contextual Decoupling*. Unconditional corpus augmentation methods (Yang et al., 2025; Ovadia et al., 2025; Tang et al., 2026) synthesize relational graphs, paraphrases, or dense instruction pairs from source documents to improve

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.

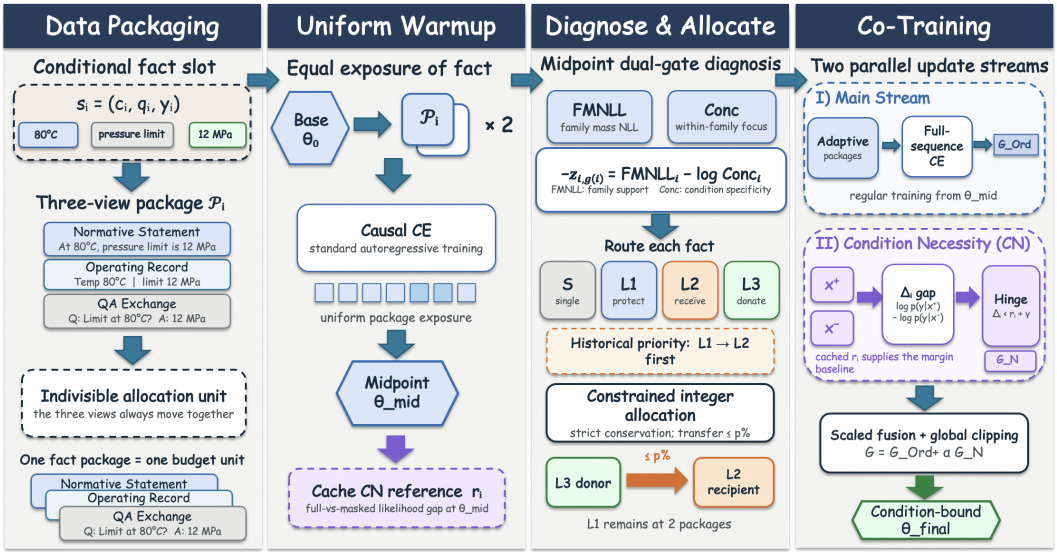


Figure 1: **Overview of DAB.** After a shared uniform stage, DAB diagnoses family support and within-family selection at the midpoint, transfers a bounded token budget from stable donors to binding-risk recipients while preserving baseline coverage, and combines full-sequence cross-entropy with an independently sampled Condition Necessity objective to enforce condition-dependent recall.

knowledge representation. However, their data exposure is determined entirely by static synthesis pipelines rather than the target model's learning dynamics; uniform token allocation inevitably wastes capacity on readily mastered facts while starving facts with acute condition–value competition. In contrast, external contextual decoupling methods (Lewis et al., 2020; Shi et al., 2024) rely on external retrieval or inference-time decoding interventions (e.g., Context-Aware Decoding) to resolve contextual ambiguities, while localized parameter-editing approaches (Meng et al., 2022; 2023) overwrite isolated associations. Yet, these methods either fail to establish persistent parametric recall in the weights or suffer from catastrophic interference when applied across thousands of interconnected domain facts.

Under a fixed training budget, conditional knowledge injection therefore requires more than passive text repetition: it demands a mechanism that dynamically assesses the model's internal memory state and reinforces condition-dependent retrieval. This naturally raises a key question: *Can fact-level memory states be diagnosed during training to adaptively steer token exposure and explicitly bind values to their applicability conditions?*

To explore this question, we introduce **DAB** (**D**iagnose, **A**llocate, and **B**ind), a framework that organizes conditional knowledge into structured fact slots, diagnoses memory bottlenecks, reallocates training exposure, and supervises condition–value bindings. Our study is organized around three central questions:

Q1: How can we diagnose whether a recall failure stems from absent candidate support or condition-specific binding confusion? We address this through ***Family-Decomposed Likelihood Diagnosis***. We organize facts sharing a normalized query key into a candidate family. DAB exactly decomposes the negative log-likelihood of the gold value into two complementary signals: FamilyMassNLL (FMNLL), measuring aggregate support for the candidate family, and Concentration (Conc), measuring the gold value's relative support among sibling candidates. Calibrated against held-out free-generation behavior, these signals route facts into **acquisition risk (L1)**, **condition-binding risk (L2)**, or **no diagnosed failure under the current gates (L3)**.

Q2: How can a fixed token budget be adaptively redistributed without compromising factual coverage? We introduce ***Coverage-Preserving Adaptive Allocation***. DAB operates across two stages under a matched ordinary-token budget. Stage I provides uniform initial exposure to

establish a stable factual baseline. At the midpoint, DAB refits diagnostic gates and executes a development-selected, dataset-specific budget transfer that reallocates packages from stable donors (L3) to binding-risk recipients (L2). Crucially, every slot retains at least one complete package and under-supported facts (L1) remain strictly protected, ensuring that adaptive training acts as targeted reinforcement rather than destructive pruning.

Q3: How can we explicitly enforce condition–value binding during optimization? We introduce ***Reference-Gated Condition Necessity (N)***. While allocation governs exposure, standard cross-entropy cannot prevent models from learning condition-independent shortcuts. N introduces an auxiliary objective contrasting the model's gold-value log-likelihood under the full condition against that under a condition-masked input. By supervising this likelihood gap relative to a slot-specific midpoint reference margin, N strengthens the binding between the correct value and its governing condition.

Experiments on GB-Standard and SQuAD-derived closed-book QA demonstrate that DAB consistently outperforms token-matched baselines across Qwen3.5-4B-Base and Llama-3.1-8B-Instruct. On SQuAD, DAB outperforms the strongest baseline, KI, by 6.22 and 19.50 percentage points, respectively. Factorial ablations and mechanistic analyses further show that adaptive allocation and Condition Necessity provide complementary benefits, simultaneously cutting sibling swaps and out-of-family errors.

Our main contributions are summarized as follows:

- **A Diagnostic Framework for Conditional Memory:** We establish an exact likelihood decomposition into family mass and within-family concentration, providing behavior-calibrated diagnostic gates that separate acquisition failure from condition–value binding failure.
- **Coverage-Preserving Token Allocation:** We formulate a two-stage training policy that transfers ordinary training tokens from stable donors to binding-risk recipients under strict budget equality, preserving factual coverage while targeting diagnosed memory bottlenecks.
- **Condition Necessity Supervision:** We propose a reference-relative margin objective that explicitly penalizes condition-independent shortcuts by contrasting full and condition-masked inputs, reinforcing parametric condition–value bindings.

2 Related Work

Continual Pre-training and Synthetic Knowledge Injection. Continual pre-training (CPT) adapts foundation models to target domains (Gururangan et al., 2020), yet factual gains remain highly non-uniform: rare recall depends heavily on pre-training frequency and diverse exposures (Allen-Zhu & Li, 2024). Synthetic injection methods mitigate surface-form memorization by restructuring text into entity graphs (EntiGraph) (Yang et al., 2025), dense instruction pairs (Knowledge-Instruct) (Ovadia et al., 2025), or prompt-engineered views (SPA) (Tang et al., 2026). However, their exposure is governed by static generation heuristics rather than model learning dynamics. In contrast, DAB diagnoses mid-training memory bottlenecks over structured fact packages, adaptively reallocating tokens while explicitly supervising condition–value bindings.

Retrieval and Parametric Knowledge Updating. Alternative paradigms supply target knowledge via external retrieval (RAG) (Lewis et al., 2020) or localized parameter editing, such as ROME (Meng et al., 2022) and MEMIT (Meng et al., 2023). However, RAG bypasses internal parametric memory, while repeated editing at scale induces severe parameter distortion and cross-fact interference (Gupta et al., 2024). Unlike surgical weight updates, DAB tackles corpus-level acquisition via standard gradient optimization, preserving guaranteed coverage for every fact while redistributing a fixed training budget across candidate families.

Conditional and Situated Knowledge. Factual correctness often depends on temporal or geographic contexts (Zhang & Choi, 2021; Dhingra et al., 2022). While conditional QA verifies explicit constraints in provided text (Lin et al., 2024a) and Context-Aware Decoding (CAD) contrasts logits at inference time (Shi et al., 2024), they fundamentally rely on test-time context. DAB instead benchmarks closed-book recall without external evidence or menus, enforcing persistent condition–value bindings directly within model parameters via a reference-relative margin loss.

Curriculum Learning and Model-Based Data Selection. Curriculum learning orders instances by difficulty (Bengio et al., 2009; Swayamdipta et al., 2020), while pretraining-scale data selection filters or weights corpora via distribution matching (DSIR) (Xie et al., 2023b), proxy mixtures (DoReMi) (Xie et al., 2023a), loss gating (Rho-1) (Lin et al., 2024b), or self-edit utility (SEAL) (Zweiger et al., 2025). Unlike document-level filtering or domain reweighting based on perplexity, DAB operates at fact-slot granularity under strict coverage preservation, transferring a bounded, development-selected budget to diagnosed binding bottlenecks.

3 Problem Formulation

3.1 Conditional Knowledge Units

We study source-grounded factual knowledge whose validity depends on an explicit applicability condition c_i (e.g., operating environments, system configurations, or temporal bounds). Each conditional fact is formulated as a slot $s_i = (c_i, q_i, y_i)$, where $q_i = (o_i, a_i, r_i, u_i)$ is a query key encoding the object, attribute, relation, and normalized unit, and y_i is the gold value. Let $\mathcal{S} = \{s_i\}_{i=1}^{N}$ be the set of validated slots. Slots sharing a normalized query key form a family: $\kappa(s_i) = \mathrm{Norm}(q_i)$, $\mathcal{F}_k = \{s_i \in \mathcal{S} : \kappa(s_i) = k\}$. The candidate set $\mathcal{V}_k = \mathrm{Unique}(\{y_i : s_i \in \mathcal{F}_k\})$ collects all distinct source-attested values within family k. Two slots are *siblings* if they share a query key but pair distinct non-empty conditions with different values. We denote family size by $n_k = |\mathcal{F}_k|$ and candidate count by $K_k = |\mathcal{V}_k|$ (with $K_i \triangleq K_{\kappa(s_i)}$); families with $K_k = 1$ are termed *singletons*.

3.2 Closed-Book Conditional Recall

Given query key q_i and condition c_i, the model must predict y_i without access to source documents or candidate menus. All evaluation question formulations are held out from training. Let $\mathcal{Q}_{\mathrm{test}}$ be the frozen benchmark questions and let $m(x;\theta) \in \{0,1\}$ indicate whether model θ answers question x correctly under deterministic normalization and exact matching.

Error Taxonomy. Predictions are categorized into three mutually exclusive outcomes: (1) **Correct**, matching y_i; (2) **Sibling Swap**, matching another family-consistent value in $\mathcal{V}_{\kappa(s_i)} \setminus \{y_i\}$; and (3) **Out-of-Family Error**, producing a value outside $\mathcal{V}_{\kappa(s_i)}$, indicating acquisition failure.

Evaluation Metric. Our primary metric is benchmark per-question accuracy:

$$\mathrm{Acc}_{\mathrm{question}}(\theta) = \frac{1}{|\mathcal{Q}_{\mathrm{test}}|} \sum_{x \in \mathcal{Q}_{\mathrm{test}}} m(x;\theta). \tag{1}$$

3.3 Learning under a Fixed Token Budget

A deterministic renderer maps each slot to three dataset-specific views, forming one indivisible fact package $\mathcal{P}_i$ with token cost b_i. An allocation policy assigns an integer package count $n_i \geq 1$ to each slot, incurring ordinary token budget $\mathcal{T}_{\mathrm{ord}}(\mathbf{n}) = \sum_i b_i n_i$.

Training comprises two stages under a matched budget $B = 2\sum_i b_i$ ($2B$ ordinary tokens total). **Stage I (Uniform Acquisition)** assigns $n_i = 2$ packages to every slot, establishing a shared baseline θ_{mid} trained via standard cross-entropy. **Stage II (Adaptive Allocation)** branches from θ_{mid}, redistributing counts $\mathbf{n}$ based on midpoint diagnostics, subject to $\sum_i b_i n_i = B$ with guaranteed coverage ($n_i \geq 1$):

$$\max_{\mathbf{n},\mathcal{L}} \ \mathrm{Acc}_{\mathrm{question}}\big(\theta_{\mathrm{final}}(\mathbf{n},\mathcal{L})\big) \qquad \text{s.t.} \quad \sum_i b_i n_i = B, \quad n_i \in \mathbb{Z}_{\geq 1}. \tag{2}$$

All diagnostic thresholds and allocation hyperparameters are calibrated exclusively on held-out development families and frozen prior to Stage II, strictly isolating the final benchmark from policy design.

4 DAB: DIAGNOSE, ALLOCATE, AND BIND

DAB coordinates three coupled operations: *Diagnose* separates weak candidate support from sibling value confusion; *Allocate* transfers a bounded token budget from stable donors to binding-risk recipients while preserving coverage; and *Bind* injects reference-gated supervision to enforce condition-dependent recall (Figure 1). Training follows the trajectory: $\theta_0 \xrightarrow{\text{Diag}} \widehat{m}^{(0)} \xrightarrow{\text{Uniform I}} \theta_{\text{mid}} \xrightarrow{\text{Diag}} \widehat{m}^{(1)} \xrightarrow{\text{Adaptive II + N}} \theta_{\text{final}}$. The midpoint state $\widehat{m}^{(1)}$ and transition history $h_i = (\widehat{m}_i^{(0)}, \widehat{m}_i^{(1)})$ govern Stage-II allocation.

4.1 DIAGNOSING CANDIDATE SUPPORT AND VALUE SELECTION

Candidate Scoring and Likelihood Decomposition. For slot s_i, diagnostic prompt x_i^{diag} specifies query key and condition without revealing candidates. For each candidate $d_{k,j} \in \mathcal{V}_k$, its teacher-forced log-likelihood is $z_{ij}^{(k)} = \sum_{t=1}^{|d_{k,j}|} \log p_\theta(d_{k,j,t} \mid x_i^{\text{diag}}, d_{k,j,<t})$. Suppressing index k, let $A_i = \log \sum_{j=1}^{K_i} \exp(z_{ij})$ and let $g(i)$ index the gold continuation. We define FamilyMassNLL (FMNLL) and Concentration (Conc) as:

$$\text{FMNLL}_i = -A_i, \qquad \text{Conc}_i = \exp\left(z_{i,g(i)} - A_i\right), \qquad \text{where} \quad -z_{i,g(i)} = \text{FMNLL}_i - \log \text{Conc}_i \,. \tag{3}$$

This exact algebraic decomposition separates aggregate family support (FMNLL) from within-family candidate competition (Conc). Because continuations omit EOS and may overlap as prefix events, FMNLL_i is treated as an operational score calibrated against behavioral outcomes rather than a normalized probability mass.

Behavior-Calibrated Routing. On development families, we calibrate decision thresholds against generation outcomes: τ_m maximizes balanced accuracy in separating out-of-family errors from family-reaching predictions, while τ_c separates sibling swaps from correct answers:

$$\widehat{m}_i = \begin{cases} S, & K_i = 1, \\ L1, & K_i > 1,\ \text{FMNLL}_i > \tau_m, \\ L2, & K_i > 1,\ \text{FMNLL}_i \le \tau_m,\ \text{Conc}_i < \tau_c, \\ L3, & \text{otherwise}, \end{cases} \tag{4}$$

where S denotes singletons, $L1$ indicates acquisition risk, $L2$ flags condition-binding risk, and $L3$ denotes no diagnosed failure under the current gates. Thresholds are recalibrated at both θ_0 and θ_{mid}.

4.2 SHARED ACQUISITION AND MIDPOINT RE-DIAGNOSIS

Stage I trains all slots uniformly ($n_i = 2$) via full-sequence cross-entropy $\mathcal{L}_{\text{CE}}(\theta) = C^{-1} \sum_{t=1}^{C} \ell_t(\theta)$. Re-diagnosing θ_{mid} yields $\widehat{m}_i^{(1)}$ and transition history $h_i = (\widehat{m}_i^{(0)}, \widehat{m}_i^{(1)})$. For binding-risk recipients ($L2$), histories distinguish emergent ($L1 \to L2$), persistent ($L2 \to L2$), and regressed ($L3 \to L2$) states. For donors ($L3$), histories distinguish stable ($L3 \to L3$) and recovered ($L2 \to L3, L1 \to L3$) states. These histories strictly rank priority during budget reallocation without altering the auxiliary schedule.

4.3 DIAGNOSIS-GUIDED ALLOCATION OF COMPLETE PACKAGES

The adaptive policy redistributes ordinary training tokens from **stable L3 donors** $\mathcal{D} = \{i : \widehat{m}_i^{(1)} = L3\}$ to **binding-risk recipients** $\mathcal{R} = \{i : \widehat{m}_i^{(1)} = L2\}$, while protecting under-supported facts and singletons $\mathcal{H} = \{i : \widehat{m}_i^{(1)} \in \{L1, S\}\}$ at $n_i = 2$. Addition and removal limits are dataset-specific; the following is the GB instantiation:

$$n_i^{\text{ada}} = \begin{cases} 2 + a_{i1} + a_{i2}, & i \in \mathcal{R}, \\ 2 - d_i, & i \in \mathcal{D}, \\ 2, & i \in \mathcal{H}, \end{cases} \qquad \text{s.t.} \quad \sum_{i \in \mathcal{R}} b_i (a_{i1} + a_{i2}) = \sum_{i \in \mathcal{D}} b_i d_i = T_{\text{move}} \le \lfloor \rho B \rfloor. \tag{5}$$

Here, $a_{i1}, a_{i2} \in \{0,1\}$ with $a_{i2} \leq a_{i1}$ denote package additions, and $d_i \in \{0,1\}$ denotes the removal of one package, ensuring that every slot retains at least one package ($n_i \geq 1$). The transfer rate ρ is selected separately for each dataset on its designated development split from a feasible dataset-specific grid. For GB, searching $\{0\%, 2\%, 4\%, 8\%, 10\%\}$ yields $\rho = 4\%$; the selected value is frozen before test evaluation (Section 5).

Because package costs vary across slots, integer reallocation maximizes the feasible transfer subject to

$$T_{\text{move}} \geq 0.99 \min \left\{ \lfloor \rho B \rfloor,\ \sum_{i \in \mathcal{D}} b_i,\ 2 \sum_{i \in \mathcal{R}} b_i \right\}.$$

Recipients are prioritized by history tier $L1 \to L2 \succ L2 \to L2 \succ L3 \to L2$, with ties broken by lower Concentration and higher FMNLL. First package increments (a_{i1}) are ranked before second increments (a_{i2}), expanding recipient coverage before assigning a fourth package. Donors are prioritized as $L3 \to L3 \succ L2 \to L3 \succ L1 \to L3$, with ties broken by lower FMNLL and higher Concentration.

4.4 Stage-II Full-Sequence Training and Condition Necessity

All Stage-II branches share full-sequence $\mathcal{L}_{\text{CE}}(\theta)$ over rendered records. To strengthen condition-dependent recall, DAB incorporates an auxiliary *Condition Necessity* (N) objective.

Reference-Gated Necessity Loss. For slot s_i, let x_i^+ denote the full prompt and x_i^- replace the condition with a "not provided" placeholder. For gold continuation $a_i^\star$, let $S_\theta(a_i^\star \mid x) = \sum_{t=1}^{|a_i^\star|} \log p_\theta(a_{i,t}^\star \mid x, a_{i,<t}^\star)$. At midpoint θ_{mid}, we cache baseline condition gap $r_i = S_{\theta_{\text{mid}}}(a_i^\star \mid x_i^+) - S_{\theta_{\text{mid}}}(a_i^\star \mid x_i^-)$. In Stage II, Condition Necessity enforces reference-relative margin $\gamma = 0.10$:

$$\mathcal{L}_{N,i}(\theta) = \left[r_i + \gamma - \left(S_\theta(a_i^\star \mid x_i^+) - \text{sg}\left(S_\theta(a_i^\star \mid x_i^-) \right) \right) \right]_+ = \left[r_i + \gamma - \Delta_i(\theta) \right]_+, \tag{6}$$

where $\Delta_i(\theta) = S_\theta(a_i^\star \mid x_i^+) - S_\theta(a_i^\star \mid x_i^-)$. The hinge remains active whenever $\Delta_i(\theta) < r_i + \gamma$, providing positive supervision $\nabla_\theta \mathcal{L}_{N,i} = -\nabla_\theta S_\theta(a_i^\star \mid x_i^+)$ until condition dependence expands by at least γ nats beyond its midpoint reference.

Gradient Control and Stream Coordination. Necessity queries follow a schedule separate from ordinary allocation. The GB configuration uses 6 evaluations per slot across Stage II. We combine ordinary gradient $\mathbf{g}_{\text{CE}} = \nabla_\theta \mathcal{L}_{\text{CE}}(\theta)$ and mean Necessity gradient $\mathbf{g}_N$ via dynamic norm scaling:

$$\mathbf{g} = \mathbf{g}_{\text{CE}} + \alpha \mathbf{g}_N, \qquad \alpha = \min\left(1,\ \frac{0.25 \|\mathbf{g}_{\text{CE}}\|_2}{\|\mathbf{g}_N\|_2 + \varepsilon}\right), \tag{7}$$

where $\varepsilon = 10^{-8}$. Across all branches, CE and CE+N variants share identical ordinary record orders, optimizer states, and update schedules. Shared Stage I, development-selected Adaptive allocation, full-sequence CE, and Condition Necessity constitute **DAB**; the GB development set selects a 4% transfer rate.

5 Experimental Setup

5.1 Corpora and Benchmarks

We evaluate DAB and baselines across two complementary benchmarks: **GB-Standard** is a Chinese corpus of national technical standards featuring semi-structured parameter tables, explicit applicability conditions, and strict operational constraints (100 standards, ∼550K source tokens). From 25,184 validated slots, we construct a closed-book benchmark of 18,706 held-out questions spanning all 100 documents. Diagnostic calibration is isolated to 2,138 development facts (4,276 questions). The two-stage ordinary training budget processes ∼14M non-padding input tokens (25.5× corpus repetition).

SQuAD (Rajpurkar et al., 2016) is a reading-comprehension benchmark constructed from Wikipedia passages. We extract 284 valid slots from the source passages and partition them into development

Table 1: Closed-book question-level accuracy (%) under matched ordinary-training-token budgets. GB contains 18,706 evaluation questions, and SQuAD contains 448 test questions. Avg. is the unweighted mean of GB and SQuAD. DAB is bold.

	Qwen3.5-4B-Base			Llama-3.1-8B-Instruct		
Model / method	GB	SQuAD	Avg.	GB	SQuAD	Avg.
Base	4.83	21.88	13.36	2.91	21.43	12.17
Raw CPT	9.40	35.49	22.45	4.02	27.01	15.52
EntiGraph	15.83	54.69	35.26	4.73	40.12	22.43
SPA	15.76	50.22	32.99	5.50	43.38	24.44
KI	23.95	57.84	40.90	11.38	50.14	30.76
DAB	**35.04**	**64.06**	**49.55**	**30.11**	**69.64**	**49.88**

and test sets, using 120 development questions to select the transfer rate ρ and 448 test questions for final reporting. Across both corpora, ordinary training budgets are strictly token-matched, and all test question formulations are held out.

5.2 Baselines and Protocol

We benchmark DAB against four representative paradigms: **Raw CPT** (Gururangan et al., 2020) pre-trains on unaugmented domain text; **EntiGraph** (Yang et al., 2025) generates relational analyses over extracted entity graphs; **SPA** (Tang et al., 2026) rewrites passages across seven prompt-engineered templates; and **Knowledge-Instruct (KI)** (Ovadia et al., 2025) synthesizes dense instruction–response pairs under supervised fine-tuning. All methods are strictly matched by realized non-padding training tokens. Controlled GB arms share identical factual assets, renderers, midpoint states, and optimizer schedules.

Evaluation is strictly closed-book without access to source passages, candidate lists, or retrieval. Predictions are scored via deterministic extraction, canonical normalization, and exact matching against ground-truth targets. Beyond overall accuracy, predictions are categorized into correct answers, sibling swaps, and out-of-family errors (Appendix A.4). We evaluate Qwen3.5-4B-Base and Llama-3.1-8B-Instruct. Fact extraction for DAB uses Qwen3.6-Plus, followed by a deterministic three-template renderer. Controlled GB branches start from identical midpoint θ_{mid}. The complete DAB corresponds to **Adaptive + CE + N**.

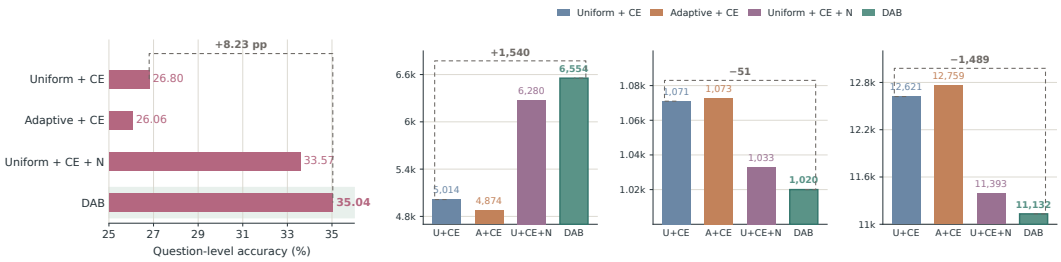
Development–test separation. For each dataset, the development split selects ρ from a feasible dataset-specific grid. On GB, searching $\{0\%, 2\%, 4\%, 8\%, 10\%\}$ yields $\rho = 4\%$. We then freeze the complete configuration before evaluation. Development and test families are disjoint, no test information is used for model or policy selection, and all reported confirmatory conclusions are based exclusively on the independent test split.

Controlled-arm notation. The labels describe Stage II: **Uniform** retains equal package counts, whereas **Adaptive** uses diagnosis-guided allocation; **CE** uses cross-entropy alone, and **CE + N** adds Condition Necessity. Thus, DAB is **Adaptive + CE + N**.

6 Main Results

6.1 End-to-End Knowledge Injection

Table 1 compares DAB with raw CPT and three representative synthetic knowledge-injection methods under matched ordinary-training-token budgets. DAB achieves the highest closed-book accuracy across both benchmarks and both backbones, as well as the best two-benchmark average for each model. On GB, it reaches 35.04% on Qwen3.5-4B-Base and 30.11% on Llama-3.1-8B-Instruct, outperforming Knowledge-Instruct, the strongest external GB baseline, by 11.09 and 18.73 percentage points, respectively. On SQuAD, DAB ranks first with **64.06%** on Qwen3.5-4B-Base and **69.64%** on Llama-3.1-8B-Instruct, outperforming the strongest baseline, Knowledge-Instruct, by **6.22** and **19.50 percentage points**, respectively.



(a) Final question-level accuracy.

(b) Final answer outcomes

Figure 2: **Controlled GB results under matched ordinary-token budgets.** (a) Final question-level accuracy for the four Allocation×Condition Necessity configurations. (b) Counts of correct answers, sibling substitutions, and out-of-family errors over the same 18,706 test questions. Dashed brackets report the difference between DAB and Uniform + CE: $+8.23$ percentage points in accuracy, $+1{,}540$ correct answers, -51 sibling substitutions, and $-1{,}489$ out-of-family errors.

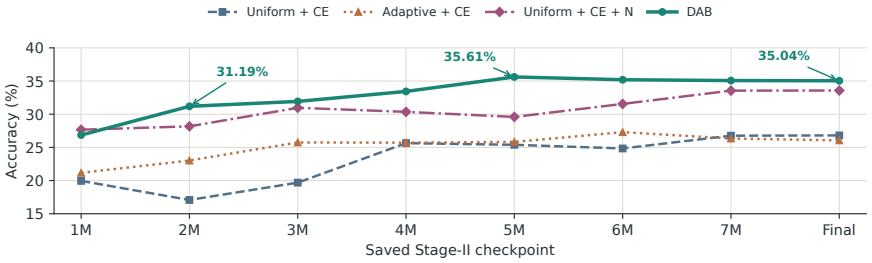


Figure 3: **Stage-II training dynamics on GB.** Accuracy of the four controlled configurations across saved checkpoints. DAB remains above Uniform + CE throughout the observed trajectory and above Uniform + CE + N from the second checkpoint onward. Final contains 7,013,204 ordinary input tokens; shared Stage-I tokens and auxiliary Condition Necessity computation are excluded.

Because all methods use the same ordinary-training-token budget, these gains demonstrate more effective use of factual supervision rather than greater training exposure. Table 1 establishes the end-to-end advantage of the complete DAB pipeline; the same-renderer comparisons in Section 6.2 separately isolate the effects of adaptive allocation and Condition Necessity.

6.2 Controlled GB Outcomes and Error Suppression

Figure 2 dissects final model predictions across the 18,706 GB questions. Under matched budgets, Uniform + CE, Adaptive + CE, Uniform + CE + N, and DAB reach 26.80%, 26.06%, 33.57%, and 35.04%, respectively. DAB generates 6,554 correct answers, 1,020 sibling swaps, and 11,132 out-of-family errors. Compared to Uniform + CE, DAB yields $+1{,}540$ **correct answers**, -51 **sibling swaps**, and $-1{,}489$ **out-of-family errors**. Holding Condition Necessity fixed, adaptive allocation adds 274 correct answers while reducing sibling swaps by 13 and out-of-family errors by 261.
Figure 3 shows that DAB's advantage is sustained across Stage-II checkpoints rather than confined to the final stopping point.

6.3 Document-Level Generalization

Evaluating performance across all 100 GB standards individually confirms that DAB's gains are broadly distributed across heterogeneous technical domains rather than concentrated in select standards (see Appendix Figure 4 for full document scatter plots). Specifically, DAB outperforms Uniform + CE on 70 of 100 standards on Qwen and 55 of 100 on Llama. Under the stricter comparison holding Condition Necessity fixed, DAB still surpasses Uniform + CE + N on 45 standards, ties on 15, and trails on 40, confirming robust cross-document efficacy across low-, medium-, and high-accuracy standards alike.

7 Ablation and Mechanistic Analysis

To dissect the internal mechanics of DAB, we investigate three fundamental questions: (1) Can diagnostic scores reliably isolate distinct failure modes? (2) Does diagnosis-guided allocation outperform unguided redistribution? (3) How do adaptive allocation and Condition Necessity interact?

7.1 Q1: Do Diagnostic Scores Reliably Isolate Failure Modes?

We evaluate diagnostic fidelity on 2,138 development slots across 587 families (Appendix Figure 6). FMNLL achieves the highest Acquisition AUROC (0.7644), significantly outperforming candidate frequency and surface length in detecting out-of-family errors. Conversely, Concentration achieves an outstanding Binding AUROC of 0.9048 in separating sibling swaps from correct answers. As shown in Appendix Figure 7, FMNLL reduces acquisition log loss by 13.99%, while Concentration provides an additional 30.38% log-loss reduction for binding prediction. ***These results support their complementary, behaviorally grounded roles in acquisition support and condition binding.***

7.2 Q2: Does Diagnosis-Guided Allocation Outperform Unguided Redistribution?

To verify whether allocation gains stem from diagnostic targeting rather than unequal exposure alone, we benchmark adaptive allocation against a diagnosis-blind *Random* control on the GB development set (2,138 facts, 4,276 questions) under identical CE objectives, total budgets, and the GB-selected transfer bound ($\rho = 4\%$). Adaptive allocation achieves **65.41% accuracy**, substantially outperforming both Uniform (62.61%) and Random (64.38%) by **+120** and **+44 correct answers**, respectively (Appendix Figure 5). This confirms that targeted redistribution guided by diagnosed memory states is essential for budget efficiency.

7.3 Q3: How Do Adaptive Allocation and Condition Necessity Interact?

Figure 2(a) presents the complete 2×2 factorial ablation on GB with Qwen3.5-4B-Base. Enabling Condition Necessity improves accuracy by 6.77 pp under Uniform allocation and by 8.98 pp under Adaptive allocation. With Condition Necessity enabled, Adaptive allocation further improves accuracy by 1.47 pp over Uniform allocation. The difference-in-differences interaction is $I = (R_{A,1} - R_{A,0}) - (R_{U,1} - R_{U,0}) =$ **2.21 percentage points**. The positive interaction ($I > 0$) demonstrates ***super-additive complementarity***: adaptive allocation channels exposure to vulnerable facts, while Condition Necessity provides the targeted gradient contrast required to anchor values to their governing conditions. Computational resource accounting is detailed in Appendix Table 4.

8 Conclusion

In this paper, we propose DAB (**D**iagnose, **A**llocate, and **B**ind), a framework that formulates conditional knowledge injection as the joint problem of diagnosing fact-level memory states, adaptively allocating training exposure, and explicitly supervising condition–value bindings. DAB outperforms token-matched baselines by up to **8.23 percentage points** in the controlled GB comparison. On SQuAD, DAB reaches **64.06%** with Qwen3.5-4B-Base and **69.64%** with Llama-3.1-8B-Instruct, outperforming the strongest baseline, KI, by **6.22** and **19.50 percentage points**, respectively.

Our analyses lead to three primary findings: (1) Decomposing gold-value negative log-likelihood into family-mass and within-family concentration terms reliably separates broad acquisition failures from fine-grained condition-binding confusions. (2) Coverage-preserving adaptive allocation channels training exposure toward vulnerable facts while retaining minimum exposure for every fact. (3) Reference-gated Condition Necessity strengthens condition-dependent recall and provides complementary gains when coupled with adaptive allocation. Together, these results demonstrate that conditional knowledge injection benefits fundamentally from ***coordinating where factual supervision is spent with how strongly recalled values are anchored to their applicability conditions***. In future work, we aim to extend DAB to multi-condition compositional reasoning and explore dynamic, streaming diagnosis during continuous pre-training.

AI USE STATEMENT

Generative AI tools were used in two roles in this work. First, Qwen3.6-Plus was used as part of the experimental data-construction pipeline to extract source-grounded conditional fact candidates from source documents and to support the construction of training data for the evaluated methods, as described in Section 5. The extracted facts used by DAB were subjected to deterministic validation procedures. Second, generative AI tools were used to edit portions of the manuscript for English clarity and readability. The authors independently verified all experimental designs, code implementations, numerical results, and scientific interpretations. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

This work does not involve human participants or private personal data collection. It evaluates closed-book factual acquisition on publicly accessible document corpora and technical standards under their applicable licenses. Models with incorrect condition–value bindings may yield inapplicable answers under safety-critical conditions; hence, model predictions should not replace authoritative technical codes or professional compliance verification.

REPRODUCIBILITY STATEMENT

We specify conditional knowledge units, evaluation tasks, metrics, and token budgets in Section 3. Section 4 provides scoring equations, routing rules, allocation policies, and Condition Necessity loss formulations. The appendix details prompt templates, hyperparameters, resource accounting, bootstrap confidence intervals, and full experimental logs.

REFERENCES

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and extraction. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 1067–1077. PMLR, 2024. URL https://proceedings.mlr.press/v235/allen-zhu24a.html.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 41–48. Association for Computing Machinery, 2009. doi: 10.1145/1553374.1553380. URL https://doi.org/10.1145/1553374.1553380.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273, 2022. doi: 10.1162/tacl_a_00459. URL https://aclanthology.org/2022.tacl-1.15/.

Akshat Gupta, Anurag Rao, and Gopala Anumanchipalli. Model editing at scale leads to gradual and catastrophic forgetting. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 15202–15232. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.902. URL https://aclanthology.org/2024.findings-acl.902/.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8360. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.740. URL https://aclanthology.org/2020.acl-main.740/.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020. URL https://papers.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Jiuheng Lin, Yuxuan Lai, and Yansong Feng. Chain of condition: Construct, verify and solve conditions for conditional question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 16596–16611. Association for Computational Linguistics, 2024a. doi: 10.18653/v1/2024.findings-emnlp.968. URL https://aclanthology.org/2024.findings-emnlp.968/.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. Not all tokens are what you need for pretraining. In *Advances in Neural Information Processing Systems*, volume 37, 2024b. doi: 10.52202/079017-0914. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/3322a9a72a1707de14badd5e552ff466-Abstract-Conference.html.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17359–17372, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=MkbcAHIYgyS.

Oded Ovadia, Meni Brief, Rachel Lemberg, and Eitam Sheetrit. Knowledge-Instruct: Effective continual pre-training from limited data using instructions. *arXiv preprint arXiv:2504.05571*, 2025. URL https://arxiv.org/abs/2504.05571.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264/.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 783–791, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-short.69. URL https://aclanthology.org/2024.naacl-short.69/.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 9275–9293. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.746. URL https://aclanthology.org/2020.emnlp-main.746/.

Kexian Tang, Jiani Wang, Shaowen Wang, and Kaifeng Lyu. SPA: A simple but tough-to-beat baseline for knowledge injection. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=wyrtuBoMkt.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S. Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. DoReMi: Optimizing data mixtures speeds up language model pretraining. In *Advances in Neural Information Processing Systems*, volume 36, pp. 69798–69818, 2023a. doi: 10.52202/075280-3059. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/dcba6be91359358c2355cd920da3fcbd-Abstract-Conference.html.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34201–34227, 2023b. doi: 10.52202/075280-1482. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/6b9aa8f418bde2840d5f4ab7a02f663b-Abstract-Conference.html.

Zitong Yang, Neil Band, Shuangping Li, Emmanuel Candès, and Tatsunori Hashimoto. Synthetic continued pretraining. In *International Conference on Learning Representations*, volume 2025, pp. 44379–44421, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/6dcf277ea32ce3288914faf369fe6de0-Abstract-Conference.html.

Michael Zhang and Eunsol Choi. SituatedQA: Incorporating extra-linguistic contexts into QA. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7371–7387, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.586. URL https://aclanthology.org/2021.emnlp-main.586/.

Adam Zweiger, Jyo Pari, Han Guo, Yoon Kim, and Pulkit Agrawal. Self-adapting language models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 82334–82365, 2025. doi: 10.52202/085713-2483. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/6b41e04c41726e2a60e456d0a2b961ab-Abstract-Conference.html.

A Experimental Details and Materials

A.1 Models and Data Generation

We conduct experiments using Qwen3.5-4B-Base and Llama-3.1-8B-Instruct as target backbones. DAB uses Qwen3.6-Plus for source-fact extraction and a deterministic three-template renderer for accepted slots.

Configuration	Value
Target models	Qwen3.5-4B-Base; Llama-3.1-8B-Instruct
Source-fact extraction	Qwen3.6-Plus
Package construction	Three deterministic templates per slot
Template views	GB: normative statement, operating record, engineer dialogue; SQuAD: natural QA, structured QA, declarative statement

Table 2: DAB models and fact-package construction.

A.2 Training Configuration

Table 3 specifies the realized Qwen GB configuration. All four controlled branches share the Uniform Stage-I endpoint, including optimizer, scheduler, and random-number states. The two-stage trajectory contains 5,368 updates, with 2,684 updates per stage.

Hyperparameter	Value
Optimizer	AdamW
Momentum coefficients (β_1, β_2)	$(0.9, 0.95)$
Numerical stabilizer ϵ	10^{-8}
Peak learning rate	10^{-5}
Learning-rate schedule	3% linear warm-up; cosine decay to 10% of peak
Weight decay	0.1 for matrix parameters; 0 for biases and lower-dimensional parameters
Maximum gradient norm	1.0
Maximum record length	512 tokens; no truncation
Ordinary microbatch	1 record; token-based update schedule
Nominal input tokens per update	$522{,}566/200 \approx 2{,}613$
N microbatch	8 slots
Training precision	FP32 parameters; BF16 autocast; TF32 enabled
Training seed	42
Total ordinary input tokens	14,026,408
N checks per slot in Stage II	6
N reference margin	0.10 natural-log units
N gradient cap	0.25 times the CE gradient norm

Table 3: Training configuration for the controlled Qwen GB experiment.

Each rendered record is encoded separately and receives one end-of-sequence token. Records are not packed into longer documents; padding is excluded from cross-entropy, and all valid causal positions receive equal weight. The optimizer and learning-rate schedule continue across the shared midpoint. The Qwen GB package-training trajectory starts from the common raw-corpus CPT checkpoint.

A.3 Token-Budget Matching

Controlled comparisons match realized non-padding input tokens after encoding with the target model's tokenizer. Complete three-view packages remain indivisible, and the allocation solver exactly matches added and removed token counts. Repeated packages count as repeated exposure to the same facts. Condition Necessity uses a separate slot-level stream, so its paired scoring workload is reported separately from the matched ordinary-training budget.

A.4 Evaluation Protocol and Prompt Templates

GB evaluation is strictly closed-book: no source passage or candidate list is supplied. The implementation builds one raw-text prompt containing three fixed demonstrations followed by the current question and answer prefix; the question is the only variable field in the template below.

GB Evaluation Prompt

请根据已经学习的国标资料直接回答下列问题。只输出答案，不要解释。把最终答案写在\boxed{} 里。

问题：已知条件为：满足“UL9”。参照《轮胎式联轴器》（GB/T 5844—2002），关于UL 型轮胎式联轴器的B，对应的规定是什么？请仅回答原文规定的数值、范围或关系符号，不要解释，也不要自行补单位。
答案：\boxed{250 mm}

问题：依据《轮胎定型硫化机》（GB/T 13579—2017），当“规格”为“890”时，调模高度是什么？请仅回答原文规定的数值、范围或关系符号，不要解释，也不要自行补单位。
答案：\boxed{200～425 mm}

问题：参照《小方头螺栓》（GB/T 35—2013），P在标准中表示什么？请仅回答原文规定内容，不要解释。
答案：\boxed{螺距}

问题：{evaluation_question}
答案：

A.4.1 GB Generation and Deterministic Scoring

Generation is greedy, with at most 64 new tokens. Prompts are left-padded, encoded without additional special tokens, and limited to 2,048 input tokens; the configured batch size is 32. Raw continuations are retained for fixed post-processing.

The extractor decodes HTML entities, removes closed reasoning blocks, applies fixed continuation delimiters, and prefers the last boxed answer with deterministic brace repair. Without a valid box, it accepts an explicit answer cue or a unique nonempty answer line. Predictions and accepted answers then receive the same Unicode, whitespace, punctuation, mathematical-surface, and relation normalization. An exact accepted answer is correct; another known family value is a sibling swap; all remaining outputs are out-of-family errors. Empty, refused, truncated, ambiguous, and unrecoverable outputs remain in the full 18,706-question denominator.

A.4.2 SQuAD Prompt and Deterministic Scoring

SQuAD Evaluation Prompt

Answer the question with only the short answer. Do not give an explanation.

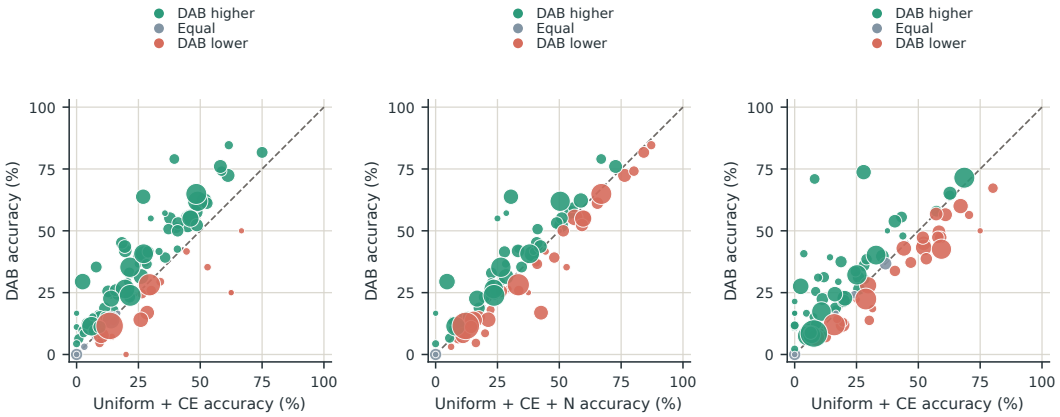
Question: {evaluation_question}
Answer:

SQuAD evaluation uses greedy decoding with at most 48 new tokens. The first nonempty continuation line is treated as the answer. Predictions and answer aliases receive deterministic Unicode, case, punctuation, and whitespace normalization before exact matching. A gold-alias match is correct, a match to another candidate in the same family is a sibling swap, and all remaining outputs are out-of-family errors.

B Supplementary Experimental Results

B.1 Document-Level Performance Across All Standards

Figure 4 plots individual standard performance across all 100 GB documents. Points above the diagonal favor DAB across low-, medium-, and high-accuracy standards.



(a) Qwen: DAB vs. Uniform + CE. **(b)** Qwen: DAB vs. Uniform + CE + N. **(c)** Llama: DAB vs. Uniform + CE.

Figure 4: **Per-standard performance across all 100 GB documents.** Each point represents one technical standard; point size is proportional to question count, and points above the diagonal favor DAB. Every panel uses the same 18,706 questions and carries its own legend.

B.2 Development Set Allocation Ablation

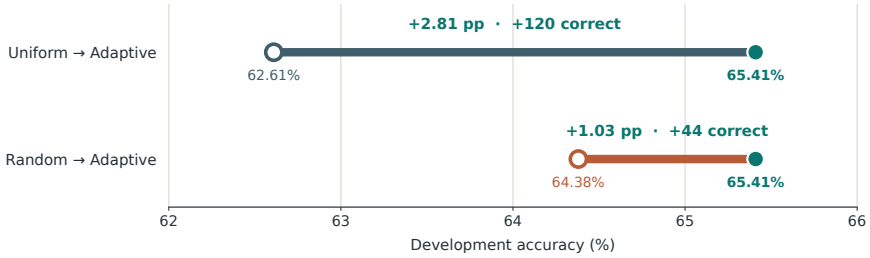


Figure 5: **Development-set allocation comparison under a fixed CE objective.** Each line compares Two-gate Adaptive with Uniform or diagnosis-blind Random over the same 2,138 facts (4,276 questions); labels report accuracy and the corresponding gains in percentage points and correct answers.

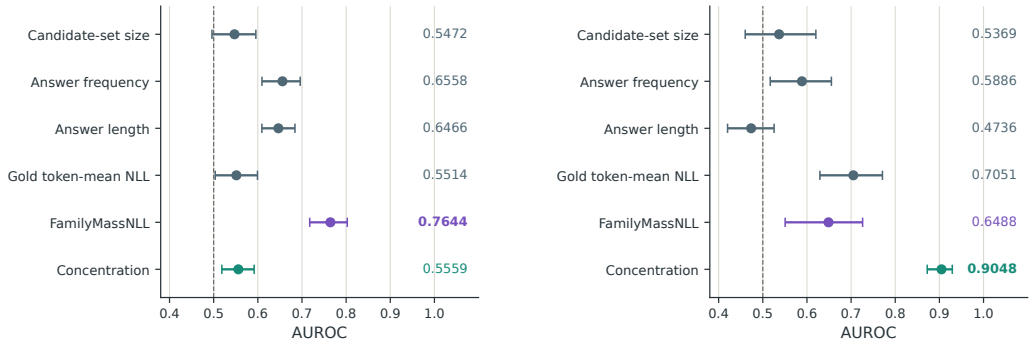
Figure 5 isolates the effect of budget allocation under a fixed CE objective on the GB development benchmark. Adaptive allocation achieves 65.41% accuracy, substantially surpassing Uniform (62.61%) and diagnosis-blind Random (64.38%) by +120 and +44 correct answers, respectively.

B.3 Computational Resource Accounting

Table 4: Computational resource accounting for controlled Qwen GB experiments.

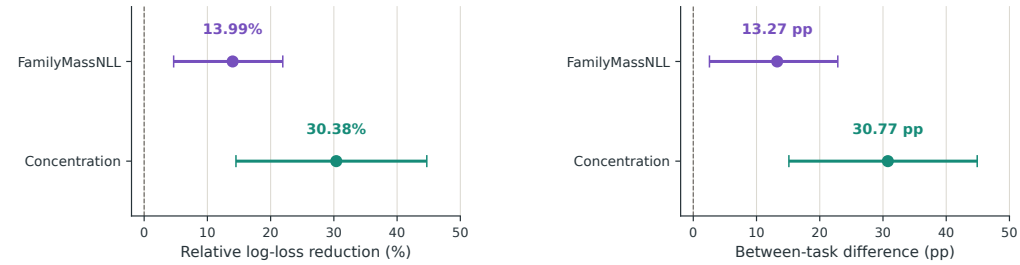
Configuration	Ordinary Tokens	Auxiliary Checks	Sequences/Check
Uniform + CE	14,026,408	0	–
Adaptive + CE	14,026,408	0	–
Uniform + CE + N	14,026,408	151,104	2
DAB	**14,026,408**	**151,104**	**2**

Table 4 details the computational workload. Every arm processes 14,026,408 ordinary tokens. Condition Necessity introduces 151,104 paired checks in Stage II.



(a) Acquisition AUROC. **(b)** Binding AUROC.

Figure 6: **Complementary diagnostic signals.** Points denote AUROC estimates, horizontal intervals denote 95% family-bootstrap confidence intervals, and the dashed line marks random ranking at 0.5.



(a) Added predictive value. **(b)** Task specificity.

Figure 7: **Distinct roles of the diagnostic signals.** FamilyMassNLL improves acquisition prediction, while Concentration adds binding information beyond FamilyMassNLL.

C Diagnostic Validation

We validate the diagnostic signals on 2,138 GB development slots from 587 families, using three free-generation probes per slot. Acquisition AUROC measures discrimination between out-of-family errors and family-reaching predictions, while Binding AUROC distinguishes sibling swaps from correct answers among family-reaching predictions. Results are aggregated across five family-held-out folds, with 95% confidence intervals obtained from 2,000 family-level bootstrap resamples.

Figure 6 shows that FMNLL provides the strongest acquisition discrimination, reaching an AUROC of 0.7644, whereas Concentration provides the strongest binding discrimination, reaching an AUROC of 0.9048. The two signals therefore capture complementary aspects of conditional recall: whether the candidate family is supported and whether the correct value is selected within that family.

Figure 7 further evaluates the signals through their added predictive value. FMNLL reduces acquisition-prediction log loss by 13.99%, while adding Concentration beyond FMNLL reduces binding-prediction log loss by 30.38%. These results support their complementary use in DAB's diagnostic routing.

D Broader Impact

DAB organizes technical knowledge into explicit condition–value associations throughout diagnosis, allocation, and training. Source-grounded fact packages preserve evidence for later review, while development-calibrated routing offers a structured workflow for adapting models to technical corpora. More accurate closed-book recall does not make the model an authoritative source: technical deployment should retain access to source evidence and appropriate human verification, particularly

where an incorrect applicability condition could affect safety or compliance. Data preparation and release must also respect source-document access and licensing requirements.