GROUP-CONSTRAINED HARDNESS REWEIGHTING FOR TOOL-CALLING AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Structured tool calls connect language-model agents to external systems, but execution succeeds only when the model decides whether to act, selects the correct schema, composes the required calls in order, and serializes valid arguments. Offline corpora combine rows that supervise different protocol roles, so a global hardness rule can change the total training weight assigned to those roles instead of isolating ambiguity among candidate schemas. We introduce PGML for method-aware tool-call supervision and HARDNESS-SCMD for group-constrained reweighting. The protocol-complete objective retains ordered targets, call counts, and valid candidate comparisons. Within candidate-one fine groups, a bounded schema-confusability-based hardness score induces a Gibbs target, contracted toward uniform only as far as weight-floor, effective-sample-size, and cap constraints require. We establish uniqueness of the Gibbs target, path-minimal contraction, and a Pythagorean bound relative to the full-space I-projection. Mean-one scaling preserves each fine group's total training weight and guarantees a global effective-sample-size lower bound. A 24-configuration diagnostic study evaluates six four-loss PGML reruns and eighteen controls under matched decoding, using coarser role-stratified weights. PGML gives competitive candidate selection on Qwen3-8B, and its SCMD branch leads multi-call count accuracy on Llama-3.2-3B, with substantial backbone-dependent selection–abstention tradeoffs.

1 INTRODUCTION

Tool calling turns a model's linguistic output into structured operations on external systems, placing API correctness at the boundary between generation and execution (Qin et al., 2024; Patil et al., 2024; Xi et al., 2025). A usable action requires the model to decide whether to invoke a tool, select the correct schema, compose the required number of calls in order, and serialize each call with valid arguments. Recent benchmarks accordingly isolate failures in AST correctness, interaction policy, turn-level calling, and selection among functionally equivalent tools (Patil et al., 2025; Yao et al., 2025; Alkhouli et al., 2025; Blankenstein et al., 2026; Lu et al., 2025). These distinctions also matter during training: rows with different protocol roles supervise different output structures, whereas candidate discrimination concerns choices within one fixed role. Offered-tool count and required-call count are therefore separate axes: a request may present several plausible tools but require one call, whereas a multi-call target primarily supervises count, ordering, and serialization.

Recent function-calling work builds supervision through verified synthesis, granular training tasks, and unified agent corpora (Liu et al., 2024; 2025b; Abdelaziz et al., 2024; Zhang et al., 2024); Hammer further shows that naming conventions and irrelevant functions can affect function selection (Lin et al., 2025). General data-selection methods concentrate optimization at token, example, and domain scales (Lin et al., 2024; Xia et al., 2024; Wang et al., 2025; Fan et al., 2024; Yu et al., 2024; 2025; Liu et al., 2025a). These two lines improve corpus quality and optimization focus, but neither separates the total training weight assigned to a protocol role from difficulty within that role. We call the latter candidate-selection hardness: difficulty in distinguishing the target schema from plausible siblings. Schema-text confusability supplies the textual-similarity component of this score. A global hardness rule can consequently shift the balance among abstention, simple calls, repeated calls, and multi-tool calls, so any gain may reflect a changed protocol mixture rather than better within-role discrimination. Our approach separates supervision design from training-weight allocation. PGML couples call generation with supervision of plan consistency, state validity, repair, and stopping.

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.

HARDNESS-SCMD then concentrates training on ambiguous candidates without changing any fine group's total training weight (figure 1).

Within each candidate-one fine group, the hardness score induces a Gibbs profile, contracted toward uniform only as far as the weight-floor, effective-sample-size, and cap constraints require. Mean-one scaling preserves the fine group's total training weight, while a single row weight multiplies the complete token, count, choice, and margin objective; non-candidate fine groups retain unit weight. The completed diagnostic study combines six four-loss PGML reruns with eighteen matched-decoding control evaluations. It identifies competitive candidate selection for Qwen3-8B's uniform PGML branch and strong multi-call count control for Llama's SCMD branch, alongside backbone- and role-dependent tradeoffs under coarser role-stratified weighting.

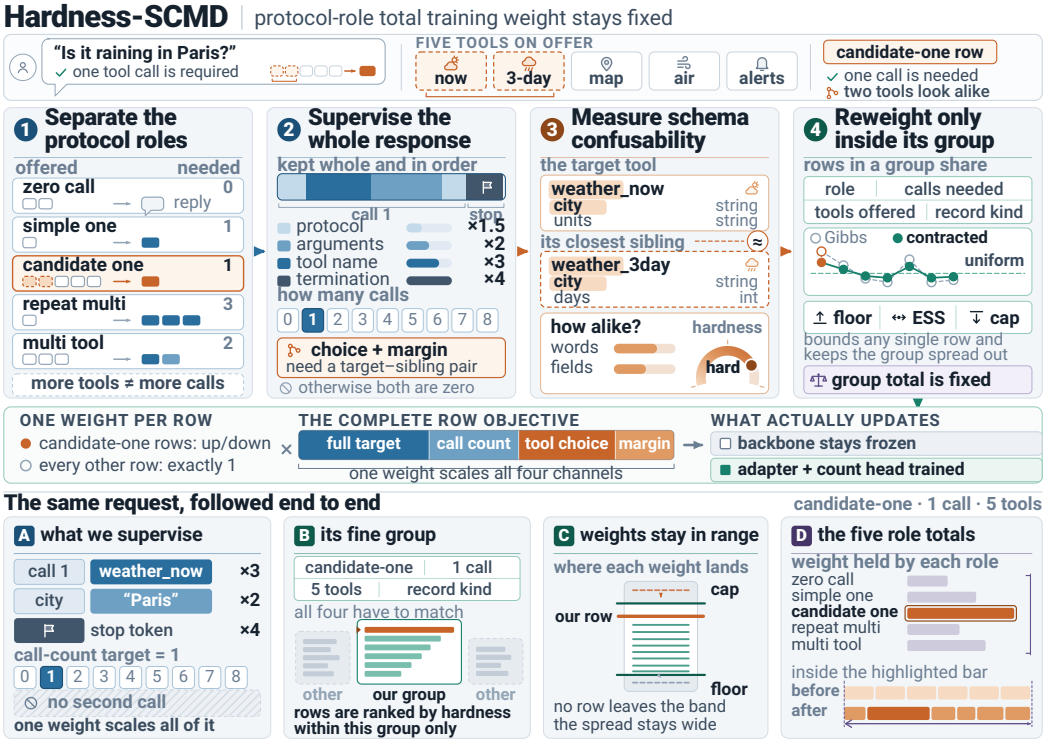

Figure 1: **HARDNESS-SCMD separates protocol-role total training weight from candidate-selection hardness.** A five-tool request requiring one call illustrates the construction. **(1)** Five protocol roles separate offered-tool count from required-call count. **(2)** The complete row objective supervises the ordered target and call count for every row, with choice and margin active only for valid target–sibling pairs. **(3)** Schema-text confusability contributes to candidate-selection hardness. **(4)** Within each fine group—defined by role, call count, tool count, and record kind—the path-minimal Gibbs-to-uniform contraction enforces the weight floor, effective-sample-size (ESS), and cap constraints while preserving total training weight. Panels A–D trace the same row: one row weight scales all four loss channels, non-candidate rows retain unit weight, and protocol-role totals remain unchanged.

Our contributions are as follows:

1. **PGML method supervision.** We couple call generation with supervision of procedural decisions and develop a protocol-complete objective that retains ordered targets and call counts, with tool-choice terms restricted to valid candidate-one pairs (section 3 and appendix B.1).
2. **Candidate-conditioned hardness.** We define a candidate-selection hardness score within candidate-one fine groups, using schema-text confusability to rank rows without conflating candidate discrimination with protocol complexity (section 3).

3. **Fine-group reweighting.** We derive the path-minimal feasible Gibbs-to-uniform contraction under weight-floor, ESS, and cap constraints, characterize its Pythagorean relation to the full-space I-projection, preserve each fine group's total training weight, and guarantee a global ESS lower bound (sections 3 and 4).
4. **Protocol-stratified study.** We evaluate six four-loss PGML reruns and eighteen controls across three backbones, identifying competitive candidate selection, improved multi-call count control, and the associated selection–abstention tradeoffs (sections 5 and 5.3).

Related work is reviewed in section 2; section 3 defines the protocol roles and fine groups and constructs the reweighting framework; section 4 proves its guarantees. Section 5 reports the diagnostic study and its results, and section 6 concludes the paper.

2 Related Work

Recent work reaches our problem through three stages: constructing reliable tool-call supervision, diagnosing structured behavior, and changing the training distribution. Each stage fixes a different part of tool-calling data: what a call should look like, how models fail, and which examples receive greater weight.

Tool-agent supervision. Toolformer learns call insertion through self-supervision (Schick et al., 2023); ToolLLM/ToolBench scales instruction and execution-path construction (Qin et al., 2024); and Gorilla studies retrieval-aware API use (Patil et al., 2024). APIGen and ToolACE verify synthesized calls through format, execution, rule, semantic, and model checks (Liu et al., 2024; 2025b), while Agent-FLAN studies how format following, reasoning, and negative examples interact during tuning (Chen et al., 2024). Granite trains seven granular function-calling tasks, and xLAM unifies heterogeneous agent data under a shared action-model pipeline (Abdelaziz et al., 2024; Zhang et al., 2024). Hammer addresses overfitting to naming conventions and sensitivity to irrelevant functions (Lin et al., 2025). This line improves the supply, coverage, and fidelity of protocol supervision.

Evaluation of structured calls. BFCL supplies AST-based checks for function calling (Patil et al., 2025); τ-bench tests policy-bound user–agent–tool interaction (Yao et al., 2025); and CONFETTI evaluates conversational calls at the turn level (Alkhouli et al., 2025). ToolSandbox adds stateful execution, implicit tool dependencies, on-policy dialogue, and insufficient-information cases (Lu et al., 2025), while BiasBusters isolates selection bias among functionally equivalent tools (Blankenstein et al., 2026). These benchmarks separate tool choice, call structure, arguments, and interaction policy. BFCL further distinguishes genuine parallel calls from its `multiple` category, which exposes several functions but usually requires one call.

Data selection and weighting. Classical importance weighting relates source and target risks under covariate shift (Sugiyama et al., 2007). Rho-1 selects token-level loss contributions (Lin et al., 2024); DSIR resamples examples (Xie et al., 2023b); and LESS and NICE estimate downstream influence (Xia et al., 2024; Wang et al., 2025). MATES and DWM make example selection or weighting responsive to the evolving model state (Yu et al., 2024; 2025). At the domain level, DoReMi and DoGE optimize mixture weights, while RegMix predicts performance across candidate mixtures (Xie et al., 2023a; Fan et al., 2024; Liu et al., 2025a). These methods determine which examples or domains receive greater weight, but they need not keep a task-defined tool-call mixture fixed.

The missing gap and our motivation. Supervision and evaluation define what a correct call should be, and weighting decides where optimization mass goes; neither line keeps protocol-role total training weight invariant while prioritizing candidate-selection hardness. A single global weight can change the balance among abstention, single-call selection, and multi-call execution before candidate discrimination improves. The remaining problem is to emphasize difficult candidate-one rows while preserving the total training weight assigned to the surrounding protocol roles. HARDNESS-SCMD addresses it with a protocol-complete loss and a fine-group Gibbs target that undergoes the smallest feasible contraction along the Gibbs-to-uniform path. Non-uniform weights remain confined to candidate-one fine groups; each fine group's total training weight stays fixed under the weight-floor, ESS, and cap constraints.

3 PROTOCOL ROLES AND GROUP-CONSTRAINED REWEIGHTING

Protocol roles distinguish the output behaviors being supervised; fine groups define where training weight may change. We construct the row objective and weights within this partition.

3.1 PROTOCOL ROLES

Let N be the number of rows, $[N] = \{1, \ldots, N\}$, and $\mathcal{D} = \{r_i\}_{i \in [N]}$. Each row $r_i = (x_i, \mathcal{T}_i, y_i, \xi_i)$ contains a request x_i, a finite schema set $\mathcal{T}_i$, the complete assistant target y_i (ordinary text or a structured continuation), and metadata ξ_i. Let $C_i = (C_{ij})_{j=1}^{k_i}$ be the ordered, possibly empty call sequence extracted from y_i, with a tool name and argument object in each C_{ij}. Then $k_i = |C_i|$ is the required call count and $a_i = |\mathcal{T}_i|$ is the number of available tools.

The classification separates function selection from call multiplicity, following BFCL's distinctions between single-function, multiple-function, parallel-call, and irrelevance scenarios (Patil et al., 2025). We define five mutually exclusive protocol roles from the reference assistant continuation: `zero_call` has $k_i = 0$; `simple_one` has $k_i = 1$ and $a_i = 1$; `candidate_one` has $k_i = 1$ and $a_i > 1$; `repeat_multi` has $k_i > 1$ with one distinct target tool; and `multi_tool` has $k_i > 1$ with at least two distinct target tools. Offered-tool count separates the single-call roles; actually invoked tool names separate the multi-call roles. The assignment uses reference calls, not predictions or hardness scores, and counts calls within one assistant continuation rather than across a dialogue.

3.2 FINE GROUPS

Define the row key $g_i = (\mathrm{role}(r_i), k_i, a_i, \mathrm{kind}(\xi_i))$ and the realized key set $\mathcal{G} = \{g_i \mid i \in [N]\}$. For $g \in \mathcal{G}$, let $G_g = \{i \in [N] \mid g_i = g\}$ and $m_g = |G_g|$; we call each G_g a fine group. Only candidate-one fine groups receive non-uniform weights, constrained to sum to m_g. This preserves the total training weight of every fine group and, after aggregation, every protocol role, while allowing emphasis within each fine group. Section 4 formalizes these guarantees.

3.3 PGML AND THE PROTOCOL-COMPLETE OBJECTIVE

PGML supplements call generation with explicit supervision of plan consistency, state validity, repair, and stopping, making procedural decisions explicit learning targets. The protocol-complete objective below adapts this principle to structured calls; appendix B.1 details the original auxiliary targets and evaluated PGML-derived lineages.

The causal LM target contains the complete assistant continuation: prompt tokens are masked, target suffixes are never truncated, and every ordered call is retained. Within a row, protocol, argument, tool-name, and termination tokens receive role-specific positive weights, normalized by their sum to give one scalar ℓ_i^{t}. We use the upright tags t, k, c, and m for the token, count, choice, and margin components.

An auxiliary predictor assigns a categorical distribution $\widehat{p}_i$ over the supported call counts, with multiclass negative log-likelihood $\ell_i^{\mathrm{k}} = -\log \widehat{p}_i(k_i)$. Let $\iota_i \in \{0, 1\}$ indicate that row i has a valid target–sibling pair with scoreable tool-name tokens on both sides. When $\iota_i = 1$, the trainer scores only the tool-name tokens of the target and selected sibling continuations. The choice component ℓ_i^{c} is their two-way cross-entropy, and the margin component ℓ_i^{m} penalizes a target–sibling score gap below $\zeta > 0$. Both components are zero when $\iota_i = 0$. The complete row objective ℓ_i is a nonnegative linear combination of the token, count, choice, and margin components, with the last two gated by ι_i. Multi-call rows therefore retain their ordered targets and count supervision without being forced into one-label candidate classification. Exact formulas, coefficients, and decoding settings appear in appendix B.

3.4 CANDIDATE-SELECTION HARDNESS SCORE

We operationalize candidate-selection hardness using schema-text confusability among sibling tools. For a tool t, let $\phi(t)$ concatenate its name, description, and parameter names, and let tok return its normalized token set. For a candidate-one row with target $t_i \in \mathcal{T}_i$ and alternatives $\mathcal{A}_i = \mathcal{T}_i \setminus \{t_i\}$,

the schema-text confusability score c_i is the maximum, over siblings in $\mathcal{A}_i$, of a convex combination of token-set Jaccard similarity and a normalized sequence-matching ratio. The exact formula is given in equation (10).

The bounded candidate-selection hardness score $s_i \in [0, 1]$ combines c_i with features that validate candidate-one membership; thus c_i is one component of s_i, not a claim of semantic equivalence. Hard-negative mining for the choice loss is a separate operation. Appendix B gives both executed constructions. Within each candidate-one fine group, s_i supplies the ranking passed to the fine-group distribution.

3.5 Constrained Gibbs-to-uniform path

The fine-group distribution converts these scores into training weights within a fine group while preserving each fine group's total training weight. For any $g \in \mathcal{G}$, let u_g be the uniform pmf on G_g, so $u_g(i) = m_g^{-1}$. For a candidate-one fine group, write $s_g = (s_i)_{i \in G_g}$. Given a temperature $T > 0$ and a contraction parameter $\delta \in [0, 1]$, define

$$q_g(i) = \frac{\exp(s_i/T)}{\sum_{j \in G_g} \exp(s_j/T)}, \qquad \pi_{g,\delta} = (1 - \delta)q_g + \delta u_g. \tag{1}$$

For any finite pmf v, define $\mathrm{ESS}(v) = (\sum_{j \in \mathrm{supp}(v)} v(j)^2)^{-1}$. The smallest δ along this path satisfying a weight-floor constraint, an ESS-ratio requirement, and a maximum normalized weight is denoted δ_g^*. Set $\pi_g = \pi_{g,\delta_g^*}$ for candidate-one fine groups and $\pi_g = u_g$ for all other fine groups. Because the fine groups partition $[N]$, $N = \sum_{g \in \mathcal{G}} m_g$.

For model parameters θ, the training weight and objective are

$$w_i = m_{g_i} \pi_{g_i}(i) \quad (i \in [N]), \qquad \mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} w_i \ell_i(\theta). \tag{2}$$

PGML and HARDNESS-SCMD thus serve complementary purposes: the former specifies supervised tool-use behavior, while the latter allocates training weight within fine groups. The protocol-complete objective retains each complete ordered target, while candidate-selection hardness scores rank rows only within candidate-one fine groups. The constrained Gibbs-to-uniform weights change relative emphasis inside those fine groups without changing any fine group's or protocol role's total training weight.

4 Properties of Group-Constrained Reweighting

We seek the smallest contraction that controls weight concentration without changing the total training weight of any fine group. Consider one nonempty candidate-one fine group G_g of size m_g. Scores are finite; $\delta \geq \delta_0$ is the minimum-mixture control that guarantees $w_i \geq \delta_0$, $\rho \in (0, 1]$ is the required ESS ratio, and $W \geq 1$ is the weight cap. Complete proofs and edge cases appear in appendix A.

4.1 Unique Gibbs target and path-minimal contraction

Before imposing feasibility, the candidate-selection hardness scores induce an entropy-regularized target. We write D_{KL} for the Kullback–Leibler divergence.

Proposition 1 (Unique fine-group Gibbs target)**.** *Let $\mathcal{F}_g(p) = \langle p, s_g \rangle - T D_{\mathrm{KL}}(p \| u_g)$ on the simplex over G_g. The distribution q_g in equation 1 is the unique maximizer and, for every competing p,*

$$\mathcal{F}_g(q_g) - \mathcal{F}_g(p) = T D_{\mathrm{KL}}(p \| q_g) \geq 0. \tag{3}$$

Along the Gibbs-to-uniform path, each control defines a lower bound on the contraction parameter. Let $\delta_{\rho,g}$ be the smallest $\delta \in [0, 1]$ for which $\mathrm{ESS}(\pi_{g,\delta})/m_g \geq \rho$, and let $\delta_{W,g}$ be the smallest δ for which $m_g \|\pi_{g,\delta}\|_\infty \leq W$. Both minima exist because the uniform endpoint satisfies the constraints under the stated assumptions.

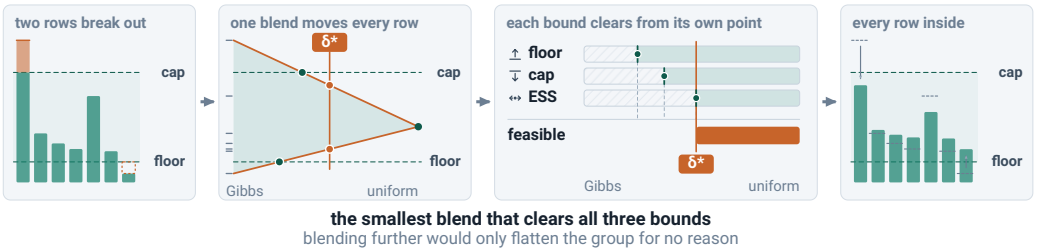

Figure 2: **Path-minimal contraction within one fine group.** *Left:* the Gibbs weights violate the floor and cap in this example. *Path:* the algorithm searches only the Gibbs-to-uniform segment in equation 1, along which every row travels a straight line to the common value, since the mixture scales each deviation from uniform by $1-\delta$; the marked crossings are where the heaviest row reaches the cap and the lightest row reaches the floor. *Thresholds:* floor, cap, and ESS each define a terminal feasible interval; their intersection begins at δ_g^*, the largest threshold in equation 4 (ESS binds here). *Right:* all constraints hold at δ_g^*, with dotted marks and arrows recording where each weight started and how far it moved; larger δ remains feasible but contracts more than required.

Theorem 1 (Path-minimal feasible contraction)**.** *The first jointly feasible point along equation 1 is*

$$\boxed{\delta_g^* = \max\{\delta_0, \delta_{\rho,g}, \delta_{W,g}\}}. \tag{4}$$

The feasible path is $[\delta_g^*, 1]$*. If* $q_g \neq u_g$*,* δ_g^* *uniquely minimizes* $D_{\mathrm{KL}}(\pi_{g,\delta}\|q_g)$ *and maximizes* $\mathcal{F}_g(\pi_{g,\delta})$ *over this path. Closed forms for the two induced thresholds appear in equation* (9)*.*

Proof sketch. Figure 2 assembles the argument. The affine mixture contracts deviations from uniform by $1-\delta$. Normalized ESS therefore increases and the largest normalized weight decreases, so the three feasible half-intervals intersect at equation 4. The Gibbs-gap identity makes divergence from q_g strictly increase when $q_g \neq u_g$; threshold derivations and edge cases appear in appendix A.

4.2 Relation to the full I-projection

The same controls admit a convex extension to the full fine-group simplex. Let $d_g = q_g - u_g$. The feasible set, its I-projection, and the path-restriction bound proved below are

$$\begin{aligned} \mathcal{C}_g = \{p \in \Delta(G_g) : \delta_0 \le m_g p(i) \le W \text{ for all } i \in G_g, \\ \mathrm{ESS}(p) \ge \rho m_g,\ \langle p - u_g, d_g\rangle \le (1-\delta_0)\|d_g\|_2^2\}, \\ p_g^{\mathrm{I}} = \arg\min_{p\in\mathcal{C}_g} D_{\mathrm{KL}}(p\|q_g), \\ D_{\mathrm{KL}}(\pi_g\|q_g) - D_{\mathrm{KL}}(p_g^{\mathrm{I}}\|q_g) \ge D_{\mathrm{KL}}(\pi_g\|p_g^{\mathrm{I}}) \ge 0. \end{aligned} \tag{5}$$

Proposition 2 (Full-space and path-restricted KL projections)**.** *For any nonempty closed convex* $\mathcal{C} \subseteq \Delta(G_g)$ *containing* u_g*, define* $p_{\mathcal{C}}^{\mathrm{I}} = \arg\min_{p\in\mathcal{C}} D_{\mathrm{KL}}(p\|q_g)$*. The path intersection is a terminal interval* $[\delta_{\mathcal{C}}, 1]$*; if* $q_g \neq u_g$*, its entry* $r_{\mathcal{C}} = \pi_{g,\delta_{\mathcal{C}}}$ *uniquely minimizes KL on that segment. The path-restriction gap obeys* $D_{\mathrm{KL}}(r_{\mathcal{C}}\|q_g) - D_{\mathrm{KL}}(p_{\mathcal{C}}^{\mathrm{I}}\|q_g) \ge D_{\mathrm{KL}}(r_{\mathcal{C}}\|p_{\mathcal{C}}^{\mathrm{I}}) \ge 0$*. Equivalently,* $\mathcal{F}_g(p_{\mathcal{C}}^{\mathrm{I}}) - \mathcal{F}_g(r_{\mathcal{C}}) \ge T D_{\mathrm{KL}}(r_{\mathcal{C}}\|p_{\mathcal{C}}^{\mathrm{I}})$*. The path-restriction gap is zero if and only if the two projections coincide. For* $\mathcal{C} = \mathcal{C}_g$ *and* $q_g \neq u_g$*,* $r_{\mathcal{C}} = \pi_g$ *and* $\delta_{\mathcal{C}} = \delta_g^*$*; if* $q_g = u_g$*, all path distributions and both projections equal* u_g*. The set* $\mathcal{C}_g$ *is nonempty, compact, and convex.*

Proof sketch. The affine preimage of $\mathcal{C}$ is a closed interval ending at 1. Closedness makes $\mathcal{C}$ compact, and continuity plus strict convexity give a unique KL projection. Because $\mathcal{C}$ contains the full-support point u_g, that projection is interior. First-order optimality and the KL three-point identity give the Pythagorean lower bound, while path monotonicity selects the entry point. The Gibbs-gap identity yields the statement for $\mathcal{F}_g$; see appendix A.

4.3 Fine-group weight preservation and global ESS lower bound

Mean-one scaling converts each fine-group distribution into training weights while preserving that fine group's total training weight.

Proposition 3 (Fine-group weight preservation and global ESS lower bound)**.** *Let $P_i = w_i/N$. The weights in equation 2 satisfy, for every $g \in \mathcal{G}$ and $i \in G_g$,*

$$\sum_{i \in G_g} w_i = m_g, \qquad \delta_0 \le w_i \le W, \qquad \mathrm{ESS}(P) \ge \rho N. \tag{6}$$

The identity also holds for unit-weight fine groups and preserves every marginal determined by the fine-group key, including each protocol role's total training weight. Combining the fine-group ESS guarantees over the partition of $[N]$ gives the global bound.

The induced gradient change also depends on the row objective (appendix A.6). We therefore apply one row weight to the complete token, count, choice, and margin objective, keeping the intervention at the row level.

Thus the construction selects the path-minimal feasible contraction, bounds the path restriction by a KL Pythagorean inequality, preserves fine-group and protocol-role total training weight, and guarantees $\mathrm{ESS}(P) \ge \rho N$.

5 Protocol-Stratified Evaluation

The completed study compares 24 configurations across three backbones under matched decoding. Six PGML-derived configurations are retrained with the four-loss objective; eighteen control adapters retain their earlier two-loss training and are re-evaluated. The empirical SCMD variant uses five protocol strata and structural difficulty, rather than the candidate-only fine groups and schema-text confusability analyzed in section 3.

5.1 Training data and comparison configurations

We use Qwen3-4B, Qwen3-8B (Qwen Team, 2025), and Llama-3.2-3B-Instruct (Meta, 2024). The shared 79,568-row pool draws on APIGen/xLAM (Liu et al., 2024), ToolACE (Liu et al., 2025b), ToolBench/ToolLLaMA G123 DFS (Qin et al., 2024), and ChainCE-derived next-step and planning records. It retains 52,176 processed original rows and adds 19,200 target-schema-only simple views and 8,192 template-generated abstention examples without benchmark content (table 3). Final-stage budgets are one epoch, 4,973 updates, effective batch size 16, seed 42, learning rate 5×10^{-6}, and rank-64 rsLoRA.

PG-U and PG-S are our PGML-derived lineages with uniform and stratified SCMD weights. Each is restarted from its original Stage-3 parent with token, count, choice, and margin coefficients 1, 0.2, 0.3, and 0.1. Choice and margin require a scoreable target–sibling pair; both vanish otherwise. Full ordered targets are preserved, with token weight 8 on the first supervised token and native assistant terminator. The generic plan, state, repair, and stop heads remain inactive. LM-U, LM-S, ToolACE*, LESS*, NICE*, and xLAM* retain their original token/count adapters; * denotes a project-adapted lineage (Liu et al., 2025b; Xia et al., 2024; Wang et al., 2025; Zhang et al., 2024). Different parent adapters and training recipes make these configuration comparisons, not isolated loss or weighting ablations.

SCMD preserves total weight within each protocol stratum, using $T = 0.7$, mixture floor 0.2, ESS ratio 0.8, and cap 3. One row weight scales every active loss. Appendix B.7 gives the pair gate, boundary supervision, and complete settings.

5.2 Evaluation data and criteria

BFCL-STATIC5 comprises five public static BFCL categories (Patil et al., 2025): Simple Python (400), Multiple (200), Parallel (200), Parallel Multiple (200), and Irrelevance (240). Simple Python and Irrelevance test simple-one and zero-call behavior; Multiple tests candidate-one selection, and Parallel tests repeated-tool calls. Parallel Multiple can require repeated or distinct-tool calls, so

Table 1: All 24 configurations (%). Loss counts active training terms (4: new PG; 2: retained controls). All/Simple/Cand./Parallel/Irrel. are BFCL AST scores; Parallel pools multi-call categories. Count_1/$\text{Count}_{\geq 2}$ measure single-/multi-call count accuracy; Over is over-call rate. Per backbone, best and second distinct values are bold and underlined, including ties. * denotes a project-adapted lineage.

Method	Loss	All ↑	Simple ↑	Cand. ↑	Parallel ↑	Irrel. ↑	Count_1 ↑	$\text{Count}_{\geq 2}$ ↑	Over ↓
Qwen3-4B									
LM-U	2	**29.8**	**0.5**	**1.5**	**87.5**	5.8	**0.8**	97.0	66.0
LM-S	2	28.4	0.0	0.0	86.0	3.3	0.0	96.3	67.3
PG-U	4	29.0	0.0	0.0	84.0	**9.6**	0.0	96.0	**65.6**
PG-S	4	28.3	0.0	0.0	85.8	3.3	0.0	96.5	66.9
ToolACE*	2	29.4	0.0	0.0	87.3	6.7	0.0	96.8	66.6
LESS*	2	28.6	0.0	0.0	87.0	2.9	0.0	**97.3**	67.2
NICE*	2	28.4	0.0	0.0	85.5	4.2	0.0	95.8	67.1
xLAM*	2	28.9	**0.5**	0.0	85.8	5.4	0.3	96.8	66.7
Qwen3-8B									
LM-U	2	**74.9**	**82.8**	**82.0**	91.0	29.2	**84.5**	97.3	**21.0**
LM-S	2	53.2	39.0	36.5	90.8	28.3	39.5	97.3	43.2
PG-U	4	63.0	72.0	65.5	89.0	2.5	72.8	95.8	31.9
PG-S	4	31.4	0.8	0.5	88.5	12.9	0.7	94.8	64.8
ToolACE*	2	46.1	38.3	4.0	89.5	22.1	28.2	97.3	49.8
LESS*	2	41.4	12.5	1.5	88.5	**44.2**	9.3	97.3	54.5
NICE*	2	69.5	74.8	63.5	**92.0**	28.3	73.3	**98.3**	26.8
xLAM*	2	31.1	0.3	0.0	90.5	9.6	0.3	96.5	65.5
Llama-3.2-3B									
LM-U	2	24.4	0.0	0.0	74.5	1.7	0.0	92.5	67.7
LM-S	2	24.4	0.3	0.0	73.8	2.5	0.2	92.8	67.3
PG-U	4	24.6	0.5	0.0	75.0	1.3	0.5	93.0	66.9
PG-S	4	25.0	0.3	0.0	75.0	3.8	0.2	**93.8**	66.8
ToolACE*	2	**29.2**	**6.0**	**10.0**	**76.8**	**4.6**	**7.8**	92.3	**62.5**
LESS*	2	24.1	0.0	1.0	73.3	1.7	0.5	92.3	67.1
NICE*	2	24.3	0.0	0.0	74.0	2.1	0.3	92.8	66.8
xLAM*	2	24.4	0.3	0.5	74.0	2.1	0.5	92.5	67.3

benchmark categories are not identical to training roles. We report BFCL AST accuracy over all 1,240 examples and by category, pooling the parallel categories over 400 examples in the main table. All 24 adapters are evaluated anew with matched greedy BF16 decoding and native stopping, without argument repair, call truncation, or count-based correction. These are public-subset diagnostic results, not full-leaderboard scores.

Count requires a valid parse and the exact reference call count; Count_0, Count_1, and $\text{Count}_{\geq 2}$ use 240, 600, and 400 references. Count_0 is strict zero-call recall, distinct from BFCL Irrelevance AST accuracy. Over-call rate uses all 1,240 references. The 24 protocol reports reuse the same 120 prediction files (29,760 responses). All checkpoints use a single training seed.

5.3 Candidate selection and multi-call control

The PGML branch is competitive in candidate selection on Qwen3-8B. PG-U correctly selects and parameterizes the tool in 131 of 200 Multiple examples (65.5%), ranking second among the eight configurations. It exceeds ToolACE*, LESS*, and xLAM* by 61.5, 64.0, and 65.5 percentage points, and NICE* by 2.0 points. Its overall AST accuracy is 62.98%, compared with 46.13%, 41.37%, and 31.13% for ToolACE*, LESS*, and xLAM*. LM-U remains strongest at 74.92% overall and 82.0% candidate accuracy; NICE* also has higher overall accuracy (69.52%). The PGML advantage is therefore clearest in candidate discrimination, rather than across every protocol role.

PG-S attains the highest multi-call count accuracy on Llama-3.2-3B: 375 of 400 multi-call references have the correct count (93.75%), the highest value among all eight configurations. This is three more than PG-U and six more than ToolACE*. Its overall AST accuracy is 25.00%, second to ToolACE*

Table 2: Configuration contrasts: S minus U, in percentage points. These comparisons retain different parent adapters and are not same-parent causal ablations. Count_0 is strict zero-call recall. Positive accuracy changes are favorable.

Backbone / lineage	Loss	ΔAST ↑	ΔCount_0 ↑	ΔCount_1 ↑	$\Delta\text{Count}_{\geq 2}$ ↑
Qwen3-4B / LM	2	-1.4	-2.9	-0.8	-0.8
Qwen3-4B / PG	4	-0.6	-7.1	0.0	+0.5
Qwen3-8B / LM	2	-21.7	-1.3	-45.0	0.0
Qwen3-8B / PG	4	-31.6	+10.4	-72.2	-1.0
Llama-3.2-3B / LM	2	0.0	0.0	+0.2	+0.3
Llama-3.2-3B / PG	4	+0.4	+0.8	-0.3	+0.8

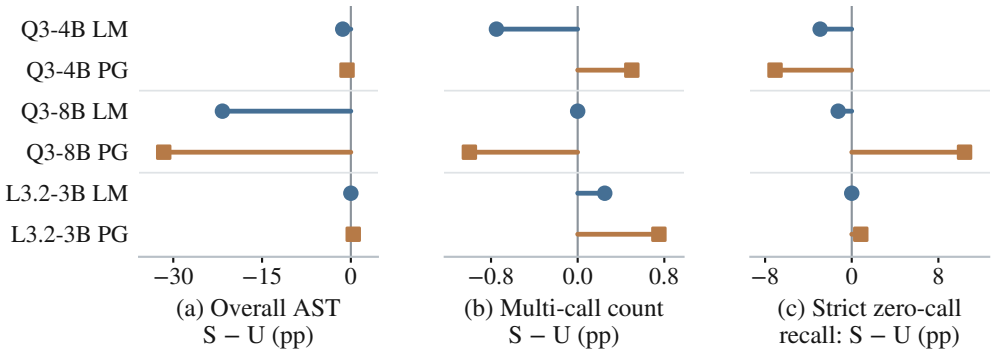

Figure 3: **How do S–U contrasts vary across protocol roles?** All six configuration differences are shown, with rightward changes indicating improvement. Llama PG-S has stronger multi-call count control, while Qwen3-8B PG-S has higher abstention recall at a substantial overall-accuracy cost.

(29.19%) and above PG-U (24.60%). Both PG configurations reach 75.0% pooled parallel AST accuracy, so the improvement concerns count control rather than a uniform increase in call correctness. On Qwen3-4B, PG-U retains the most correct abstention decisions, with strict zero-call recall of $22/240$ (9.17%), versus $15/240$ (6.25%) for the next-best ToolACE*.

5.4 Role-specific configuration contrasts

Table 2 and figure 3 report every S–U contrast. On Qwen3-8B, PG-S has higher strict zero-call recall than PG-U, 11.67% versus 1.25%, yielding 25 more correct abstentions. This accompanies a large selection cost: Count_1 falls from 72.83% to 0.67%, and overall AST accuracy falls from 62.98% to 31.37%. On Qwen3-4B, both PG variants have zero Count_1, and PG-S scores 28.31% overall versus PG-U's 28.95%. The small Llama gain and the Qwen reversals show why configurations must be evaluated by role and backbone.

Across all configurations, multi-call count accuracy remains 92.25–98.25%, yet single-call behavior varies sharply (appendix C). The four-loss PGML configurations are competitive in candidate selection and count control on different backbones, but preserved training weight alone does not guarantee the desired protocol behavior.

6 Conclusion

PGML supplies method-level tool-call supervision, while HARDNESS-SCMD controls candidate emphasis without changing protocol-role training weight. Its path-minimal contraction preserves fine-group totals and admits global ESS and KL guarantees. The completed diagnostic study finds competitive candidate selection for Qwen3-8B's PGML branch and the strongest multi-call count control for Llama's SCMD branch, alongside substantial selection–abstention tradeoffs. Role-specific evaluation connects these gains and tradeoffs to the behaviors supervised by PGML.

AI USE STATEMENT

Generative AI tools were used for manuscript editing and formatting. The authors reviewed all AI-assisted content and take full responsibility for the final work.

REFERENCES

Ibrahim Abdelaziz, Kinjal Basu, Mayank Agarwal, Sadhana Kumaravel, Matthew Stallone, Rameswar Panda, Yara Rizk, G P Shrivatsa Bhargav, Maxwell Crouse, Chulaka Gunasekara, Shajith Ikbal, Sachindra Joshi, Hima Karanam, Vineet Kumar, Asim Munawar, Sumit Neelam, Dinesh Raghu, Udit Sharma, Adriana Meza Soria, Dheeraj Sreedhar, Praveen Venkateswaran, Merve Unuvar, David Daniel Cox, Salim Roukos, Luis A. Lastras, and Pavan Kapanipathi. Granite-function calling model: Introducing function calling abilities via multi-task learning of granular tasks. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 1131–1139. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-industry.85. URL https://aclanthology.org/2024.emnlp-industry.85/.

Tamer Alkhouli, Katerina Margatina, James Gung, Raphael Shu, Claudia Zaghi, Monica Sunkara, and Yi Zhang. CONFETTI: Conversational function-calling evaluation through turn-level interactions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7993–8006, 2025. doi: 10.18653/v1/2025.acl-long.394. URL https://aclanthology.org/2025.acl-long.394/.

Thierry Blankenstein, Jialin Yu, Zixuan Li, Vassilis Plachouras, Sunando Sengupta, Philip Torr, Yarin Gal, Alasdair Paren, and Adel Bibi. BiasBusters: Uncovering and mitigating tool selection bias in large language models. In *International Conference on Learning Representations*, pp. 102460–102489, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/a79875cc0d046ce7ce65f03f3affaa9e-Paper-Conference.pdf.

Zehui Chen, Kuikun Liu, Qiuchen Wang, Wenwei Zhang, Jiangning Liu, Dahua Lin, Kai Chen, and Feng Zhao. Agent-FLAN: Designing data and methods of effective agent tuning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 9354–9366. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.557. URL https://aclanthology.org/2024.findings-acl.557/.

Simin Fan, Matteo Pagliardini, and Martin Jaggi. DOGE: Domain reweighting with generalization estimation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12895–12915, 2024. URL https://proceedings.mlr.press/v235/fan24e.html.

Qiqiang Lin, Muning Wen, Qiuying Peng, Guanyu Nie, Junwei Liao, Xiaoyun Mo, Jiamu Zhou, Cheng Cheng, Yin Zhao, Jun Wang, and Weinan Zhang. Robust function-calling for on-device language model via function masking. In *International Conference on Learning Representations*, pp. 85446–85466, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/d45e0bfb5a39477d56b55c0824200008-Paper-Conference.pdf.

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, pp. 29029–29063, 2024. doi: 10.52202/079017-0914. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/3322a9a72a1707de14badd5e552ff466-Paper-Conference.pdf.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. RegMix: Data mixture as regression for language model pre-training. In *International Conference on Learning Representations*, pp. 38305–38339, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/5f67d864aae6115374fed7beddd119e0-Paper-Conference.pdf.

Weiwen Liu, Xu Huang, Xingshan Zeng, Xinlong Hao, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Zhengying Liu, Yuanqing Yu, Zezhong Wang, Yuxian Wang, Wu Ning, Yutai Hou, Bin Wang, Chuhan Wu, Xinzhi Wang, Yong Liu, Yasheng Wang, Duyu Tang, Dandan Tu, Lifeng Shang, Xin Jiang, Ruiming Tang, Defu Lian, Qun Liu, and Enhong Chen. ToolACE: Winning the points of LLM function calling. In *International Conference on Learning Representations*, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/663865ea167425c6c562cb0b6bcf76c7-Abstract-Conference.html.

Zuxin Liu, Thai Hoang, Jianguo Zhang, Ming Zhu, Tian Lan, Shirley Kokane, Juntao Tan, Weiran Yao, Zhiwei Liu, Yihao Feng, Rithesh Murthy, Liangwei Yang, Silvio Savarese, Juan Carlos Niebles, Huan Wang, Shelby Heinecke, and Caiming Xiong. APIGen: Automated pipeline for generating verifiable and diverse function-calling datasets. In *Advances in Neural Information Processing Systems*, volume 37, pp. 54463–54482, 2024. doi: 10.52202/079017-1725. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/61cce86d180b1184949e58939c4f983d-Paper-Datasets_and_Benchmarks_Track.pdf.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. ToolSandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1160–1183. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.65. URL https://aclanthology.org/2025.findings-naacl.65/.

Meta. Llama 3.2 3b model card. Official model card, September 2024. URL https://huggingface.co/meta-llama/Llama-3.2-3B. Version release date: September 25, 2024.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive APIs. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-4020. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/e4c61f578ff07830f5c37378dd3ecb0d-Abstract-Conference.html.

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 48371–48392. PMLR, 2025. URL https://proceedings.mlr.press/v267/patil25a.html.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=dHng2O0Jjr.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. doi: 10.48550/arXiv.2505.09388. URL https://arxiv.org/abs/2505.09388.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-2997. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.

Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(35):985–1005, 2007. URL https://www.jmlr.org/papers/v8/sugiyama07a.html.

Jingtan Wang, Xiaoqiang Lin, Rui Qiao, Pang Wei Koh, Chuan-Sheng Foo, and Bryan Kian Hsiang Low. NICE data selection for instruction tuning in LLMs with non-differentiable evaluation

metric. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 63662–63689. PMLR, 2025. URL https://proceedings.mlr.press/v267/wang25bm.html.

Zhiheng Xi, Yiwen Ding, Wenxiang Chen, Boyang Hong, Honglin Guo, Junzhe Wang, Xin Guo, Dingwen Yang, Chenyang Liao, Wei He, Songyang Gao, Lu Chen, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, Xipeng Qiu, Xuanjing Huang, Zuxuan Wu, and Yu-Gang Jiang. AgentGym: Evaluating and training large language model-based agents across diverse environments. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27914–27961. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1355. URL https://aclanthology.org/2025.acl-long.1355/.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54104–54132. PMLR, 2024. URL https://proceedings.mlr.press/v235/xia24c.html.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy 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, 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, 2023b. doi: 10.52202/075280-1482. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/6b9aa8f418bde2840d5f4ab7a02f663b-Abstract-Conference.html.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations*, pp. 9965–10017, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/1b126cc38b8638e07bef37e7b2bb72bf-Paper-Conference.pdf.

Yang Yu, Kai Han, Hang Zhou, Yehui Tang, Kaiqi Huang, Yunhe Wang, and Dacheng Tao. LLM data selection and utilization via dynamic bi-level optimization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 72995–73008. PMLR, 2025. URL https://proceedings.mlr.press/v267/yu25g.html.

Zichun Yu, Spandan Das, and Chenyan Xiong. MATES: Model-aware data selection for efficient pretraining with data influence models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 108735–108759. Curran Associates, Inc., 2024. doi: 10.52202/079017-3453. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/c4bec0d2fd217e6c2c3eafeced432582-Paper-Conference.pdf.

Jianguo Zhang, Tian Lan, Ming Zhu, Zuxin Liu, Thai Hoang, Shirley Kokane, Weiran Yao, Juntao Tan, Akshara Prabhakar, Haolin Chen, Zhiwei Liu, Yihao Feng, Tulika Awalgaonkar, Rithesh Murthy, Eric Hu, Zeyuan Chen, Ran Xu, Juan Carlos Niebles, Shelby Heinecke, Huan Wang, Silvio Savarese, and Caiming Xiong. xLAM: A family of large action models to empower AI agent systems. *arXiv preprint arXiv:2409.03215*, 2024. doi: 10.48550/arXiv.2409.03215. URL https://arxiv.org/abs/2409.03215.

A Proofs and Additional Guarantees

The first three subsections concern one nonempty fine group G_g of size m_g. We abbreviate $q = q_g$ and $u = u_g$, and write $p(i)$ for coordinate i of a fine-group distribution p.

A.1 Fine-group Gibbs identity

The uniform-prior Gibbs target satisfies

$$Z = \sum_{j \in G_g} \exp(s_j/T), \qquad \log q(i) = s_i/T - \log Z, \quad i \in G_g.$$

For any distribution p on G_g,

$$TD_{\mathrm{KL}}(p\|q) = T \sum_{i \in G_g} p(i) \log p(i) - \sum_{i \in G_g} p(i) s_i + T \log Z.$$

Because $D_{\mathrm{KL}}(p\|u) = \sum_{i \in G_g} p(i) \log p(i) + \log m_g$, rearrangement gives

$$\mathcal{F}_g(p) = T \log Z - T \log m_g - TD_{\mathrm{KL}}(p\|q). \tag{7}$$

At $p = q$ the KL term vanishes. KL nonnegativity proves equation 3, and equality only at $p = q$ proves uniqueness.

A.2 Uniform contraction, ESS, and maximum weight

For $\pi_\delta = (1-\delta)q + \delta u$,

$$\pi_\delta(i) \geq \delta/m_g \quad (i \in G_g), \qquad \pi_\delta - u = (1-\delta)(q-u).$$

Expanding the squared coordinate sum and using $\sum_{i \in G_g} q(i) = 1$ yields

$$m_g \sum_{i \in G_g} \pi_\delta(i)^2 = 1 + (1-\delta)^2 D_g, \qquad D_g = m_g \sum_{i \in G_g} q(i)^2 - 1.$$

Therefore

$$\frac{\mathrm{ESS}(\pi_\delta)}{m_g} = \left[1 + (1-\delta)^2 D_g\right]^{-1}. \tag{8}$$

The mixture applies the same affine map to every coordinate. With $Q_g = \max_{i \in G_g} q(i) = \|q\|_\infty$ and $\kappa_g = W/m_g$,

$$\max_{i \in G_g} \pi_\delta(i) = m_g^{-1} + (1-\delta)(Q_g - m_g^{-1}).$$

Solving the ESS-ratio and cap constraints gives

$$\delta_{\rho,g} = \begin{cases} 0, & D_g = 0, \\ \left[1 - \sqrt{(1-\rho)/(\rho D_g)}\right]_+, & D_g > 0, \end{cases} \quad \delta_{W,g} = \begin{cases} 0, & Q_g \leq \kappa_g, \\ \dfrac{Q_g - \kappa_g}{Q_g - 1/m_g}, & Q_g > \kappa_g. \end{cases} \tag{9}$$

The mixture floor requires $\delta \geq \delta_0$. Intersecting this half-interval with the ESS and cap half-intervals proves equation 4. The cap is feasible at uniform whenever $W \geq 1$.

A.3 Path-restricted projection

Let $K(\delta) = D_{\mathrm{KL}}(\pi_\delta\|q)$. Differentiating along the path gives

$$K'(0^+) = 0, \qquad K''(\delta) = \sum_{i \in G_g} \frac{(u(i) - q(i))^2}{\pi_\delta(i)} \geq 0, \qquad \delta \in (0, 1].$$

Hence K is nondecreasing on $[0, 1]$ and strictly increasing on $(0, 1]$ when $q \neq u$. The smallest feasible contraction minimizes K. Equation (7) also gives

$$\mathcal{F}_g(q) - \mathcal{F}_g(\pi_\delta) = TK(\delta),$$

so the same point maximizes the entropy-regularized score along the feasible segment. This proves theorem 1.

A.4 General convex feasible sets and I-projection

Let $\mathcal{C} \subseteq \Delta(G_g)$ be nonempty, closed, and convex with $u \in \mathcal{C}$, and define $I_{\mathcal{C}} = \{\delta \in [0,1] : \pi_\delta \in \mathcal{C}\}$. The path map is affine and continuous, so $I_{\mathcal{C}}$ is nonempty, closed, and convex. Because $1 \in I_{\mathcal{C}}$, it equals $[\delta_{\mathcal{C}}, 1]$ for a smallest $\delta_{\mathcal{C}}$.

Finite scores give $q(i) > 0$ for every coordinate, so $D_{\mathrm{KL}}(\cdot\|q)$ is continuous and strictly convex on the simplex. Since $\mathcal{C}$ is a closed subset of the compact simplex, a minimizer exists and strict convexity makes it unique. This I-projection $p^{\mathrm{I}}_{\mathcal{C}}$ has full support: if one of its coordinates were zero, moving toward the feasible full-support point u would have directional derivative $-\infty$. Forward KL is therefore differentiable at the projection. For any $r \in \mathcal{C}$, first-order optimality and the KL three-point identity give

$$\begin{aligned} &D_{\mathrm{KL}}(r\|q) - D_{\mathrm{KL}}(p^{\mathrm{I}}_{\mathcal{C}}\|q) - D_{\mathrm{KL}}(r\|p^{\mathrm{I}}_{\mathcal{C}}) \\ &\quad = \left\langle r - p^{\mathrm{I}}_{\mathcal{C}}, \log \frac{p^{\mathrm{I}}_{\mathcal{C}}}{q} \right\rangle \geq 0. \end{aligned}$$

Taking $r = \pi_{\delta_{\mathcal{C}}}$ proves the Pythagorean bound; the monotonicity of K makes this r the unique path-restricted projection when $q \neq u$. Because the gap is at least $D_{\mathrm{KL}}(r\|p^{\mathrm{I}}_{\mathcal{C}})$, KL separation shows that the gap vanishes if and only if the projections coincide. By equation (7), the same inequality is equivalent to $\mathcal{F}_g(p^{\mathrm{I}}_{\mathcal{C}}) - \mathcal{F}_g(r) \geq T D_{\mathrm{KL}}(r\|p^{\mathrm{I}}_{\mathcal{C}})$.

For the set $\mathcal{C}_g$ in equation (5), the uniform distribution is feasible. The ESS constraint is equivalent to the convex sublevel set $\sum_i p(i)^2 \leq 1/(\rho m_g)$; the coordinate bounds and directional constraint are closed halfspaces. Thus $\mathcal{C}_g$ is a nonempty closed convex subset of the compact simplex. If $d_g = q_g - u_g \neq 0$, then $\langle \pi_{g,\delta} - u_g, d_g \rangle = (1-\delta)\|d_g\|_2^2$, so the directional constraint is equivalent to $\delta \geq \delta_0$. That condition implies the coordinate floor along the path, and the ESS and cap constraints contribute $\delta_{\rho,g}$ and $\delta_{W,g}$; hence the entry parameter is δ^*_g. If $d_g = 0$, then $I_{\mathcal{C}_g} = [0,1]$ and $\delta_{\mathcal{C}_g} = 0$, while the algorithm may designate $\delta^*_g = \delta_0$; every parameter represents the same distribution u_g. This proves proposition 2.

A.5 Fine-group weight preservation and global ESS

For every candidate-one fine group,

$$\sum_{i \in G_g} w_i = m_g \sum_{i \in G_g} \pi_g(i) = m_g.$$

Unit-weight non-candidate fine groups obey the same identity. If a statistic $f_i = f(g_i)$ is constant on each fine group, then

$$\sum_{i=1}^{N} w_i f_i = \sum_{g \in \mathcal{G}} f(g) \sum_{i \in G_g} w_i = \sum_{g \in \mathcal{G}} f(g) m_g = \sum_{i=1}^{N} f_i.$$

This proves preservation of every marginal encoded by the fine-group key. For $i \in G_g$, the mixture floor gives $w_i = m_g \pi_g(i) \geq \delta_0$, and the cap gives $w_i \leq W$.

For the global row distribution $P = (P_i)_{i=1}^N$, where $P_i = w_i/N$, the fine-group ESS guarantees imply

$$\sum_{i=1}^{N} P_i^2 = \frac{1}{N^2} \sum_{g \in \mathcal{G}} m_g^2 \sum_{i \in G_g} \pi_g(i)^2 \leq \frac{1}{N^2} \sum_{g \in \mathcal{G}} \frac{m_g}{\rho} = \frac{1}{\rho N}.$$

Taking the reciprocal proves $\mathrm{ESS}(P) \geq \rho N$ and completes proposition 3.

A.6 Loss–weight interaction

For any loss channel A, let $\mathcal{R}^{(A)}_\mu(\theta) = \sum_{i=1}^N \mu_i \ell_i^{(A)}(\theta)$ and $\gamma_i^{(A)} = \nabla_\theta \ell_i^{(A)}(\theta)$. Linearity gives

$$\nabla_\theta \mathcal{R}^{(A)}_\mu - \nabla_\theta \mathcal{R}^{(A)}_\nu = \sum_{i=1}^{N} (\mu_i - \nu_i) \gamma_i^{(A)}.$$

For channels A, B and row distributions $\mu, \nu \in \Delta^{N-1}$, subtraction yields

$$(\nabla_\theta \mathcal{R}_\mu^{(A)} - \nabla_\theta \mathcal{R}_\nu^{(A)}) - (\nabla_\theta \mathcal{R}_\mu^{(B)} - \nabla_\theta \mathcal{R}_\nu^{(B)}) = \sum_{i=1}^{N} (\mu_i - \nu_i)(\gamma_i^{(A)} - \gamma_i^{(B)}).$$

Fix channel A and assume $\|\gamma_i^{(A)}\|_2 \le M$ for every $i \in [N]$ and some $M \ge 0$. If $U \sim \mathrm{Unif}([N])$ and $\widehat{\gamma}^{(A)} = w_U \gamma_U^{(A)}$, then

$$\mathbb{E}_U[\widehat{\gamma}^{(A)}] = \frac{1}{N} \sum_{i=1}^{N} w_i \gamma_i^{(A)}, \qquad \mathbb{E}_U[\|\widehat{\gamma}^{(A)}\|_2^2] \le \frac{M^2}{N} \sum_{i=1}^{N} w_i^2, \qquad \|\widehat{\gamma}^{(A)}\|_2 \le WM.$$

The ESS and cap therefore control local stochastic-gradient moments and range for the chosen row distribution.

A.7 Edge cases

Empty fine groups are omitted. If $q = u$, the fine group is already uniform and every contraction value denotes the same distribution. If $W < 1$, even the uniform mean-one weights violate the cap, so the configuration is rejected. Non-candidate fine groups use unit weights rather than zero weight; the candidate-only profile changes relative effort only inside its declared fine groups.

B Implementation and Evaluation Details

We first describe PGML's method-supervision design, then specify the candidate-focused four-loss construction. The numerical study in section 5 evaluates six four-loss PGML reruns and eighteen two-loss controls with coarser role-stratified weights (appendix B.7).

B.1 PGML method supervision

PGML is our supervision component; its original training implementation is named PGDML-G. It augments call generation with auxiliary targets derived from annotated tool-use records. Plan targets encode the recorded plan-consistency judgment. State targets use tool and argument validity when an observation is available; unavailable observation-dependent targets are masked. Repair and stopping targets distinguish cases requiring correction or further action from those in which the procedure is complete. A soft target over continue, finish, and repair further supervises the stopping decision. These signals are training targets, not a guarantee of valid online execution; verifier-derived labels likewise do not constitute formal proofs.

This supervision addresses a limitation of token-level imitation: matching a call sequence does not explicitly distinguish why the agent should continue, repair, or stop. The protocol-complete adaptation in section 3 retains the full ordered continuation and supervises call count on every row, while restricting choice and margin terms to valid candidate comparisons. PGML specifies these learning targets; SCMD changes the relative training weight assigned to rows. The weighting analysis applies to the stated complete row loss and does not require the original PGDML-G auxiliary heads.

The PG-U and PG-S configurations descend from our PGML training branch, so they represent the supervision side of the proposed approach rather than an external baseline method. Their new final-stage continuation enables token, count, choice, and margin losses and strengthens boundary-token supervision. The earlier plan, state, repair, and stop heads remain inactive; the comparison does not isolate any individual auxiliary target.

B.2 Protocol strata and target preservation

The preprocessing path reads the complete supervised call list rather than the first call. For native conversations it selects the assistant turn containing the richest valid call list, breaking an exact tie in favor of the later turn. An explicit ordinary assistant response is a zero-call target. The declared call count must equal the rendered count and must lie in $\{0, \ldots, 8\}$.

Tokenization locates every rendered name and argument span in target order, including repeated uses of the same function. Prompt context may be compacted to meet the sequence budget, but the independently rendered assistant suffix must remain token-identical and is never truncated. Rows whose target exceeds the configured target budget fail preprocessing rather than receiving partial supervision.

Table 3 gives the source composition of the protocol-complete pool used for continuation. The source names refer to processed records, not complete public releases.

Table 3: Training-pool sources. All 52,176 original IDs and targets are retained; derived simple views and template-generated zero-call records bring the total to 79,568.

Component	Rows
APIGen/xLAM function calling	29,206
ToolACE dialogues	3,770
ToolBench/ToolLLaMA G123 DFS	6,107
ChainCE-derived next-step supervision	7,705
ChainCE-derived planning supervision	5,388
Original-row subtotal	52,176
Candidate-to-simple views	19,200
Benchmark-free zero-call rows	8,192
Total	79,568

The public source families are documented in the APIGen/xLAM release, ToolACE release, and ToolBench repository. ChainCE denotes the next-step and planning source labels retained in the project's preprocessing records. The 32,976 APIGen/xLAM and ToolACE rows supply multi-call targets; the remaining 19,200 original rows supply candidate-one targets. Each simple view retains only its parent's target schema. Zero-call rows are generated deterministically from ordinary language-task templates with irrelevant tools; neither their requests nor their targets are drawn from BFCL. The pool contains 8,192 zero-call, 19,200 simple-one, 19,200 candidate-one, 12,185 repeat-multi, and 20,791 multi-tool rows.

For the fixed BFCL subset, official category labels are retained at evaluation time. Parallel Multiple contains 9 repeat-multi and 191 multi-tool reference targets, illustrating why its category name is not a synonym for the multi-tool training role. Multi-call AST evaluation checks the benchmark's call-matching criterion, rather than certifying the order of online execution.

B.3 Exact row objective

Let S_i^+ and S_i^- be the mean tool-name log probabilities of the target and selected sibling continuations. With $[z]_+ = \max\{z, 0\}$ and margin $\zeta > 0$, the candidate channels are

$$\ell_i^{\mathrm{c}} = -\log \frac{\exp(S_i^+)}{\exp(S_i^+) + \exp(S_i^-)}, \qquad \ell_i^{\mathrm{m}} = [\zeta - S_i^+ + S_i^-]_+.$$

They are zero when $\iota_i = 0$. For coefficients $\lambda_{\mathrm{t}} > 0$ and $\lambda_{\mathrm{k}}, \lambda_{\mathrm{c}}, \lambda_{\mathrm{m}} \geq 0$, the complete row objective is

$$\ell_i = \lambda_{\mathrm{t}} \ell_i^{\mathrm{t}} + \lambda_{\mathrm{k}} \ell_i^{\mathrm{k}} + \iota_i \big(\lambda_{\mathrm{c}} \ell_i^{\mathrm{c}} + \lambda_{\mathrm{m}} \ell_i^{\mathrm{m}}\big).$$

B.4 Exact candidate-selection hardness score

For a tool t, let $\phi(t)$ concatenate its name, description, and parameter names, and let tok return its normalized token set. For target t_i and sibling set $\mathcal{A}_i$, define the schema-text confusability score

$$c_i = \max_{t \in \mathcal{A}_i} \left[\alpha\, J(\mathrm{tok}(\phi(t_i)), \mathrm{tok}(\phi(t))) + (1 - \alpha)\, R(\phi(t_i), \phi(t))\right], \qquad \alpha \in [0, 1], \tag{10}$$

where J is token-set Jaccard similarity and R is a normalized sequence-matching ratio.

Let $z_i^{\mathrm{a}} = \log a_i / \log 17$, let $z_i^{\mathrm{u}} = \min\{|\{\mathrm{name}(C_{ij}) \mid 1 \leq j \leq k_i\}|, 8\}/8$, and let $z_i^{\mathrm{r}} \in \{0, 1\}$ indicate separated repeated calls to one function. The name-validity factor η_i^{v} is one for a valid target

and 0.05 otherwise; the duplicate factor η_i^{d} is one unless the target repeats an exact call, in which case it is 0.10. Define $\mathrm{clip}_{[0,1]}(x) = \min\{1, \max\{0, x\}\}$. With c_i instantiated at $\alpha = 0.55$, the builder computes

$$\beta_i = \mathrm{clip}_{[0,1]}\left(0.35 z_i^{\mathrm{a}} + 0.40 c_i + 0.15 z_i^{\mathrm{u}} + 0.10 z_i^{\mathrm{r}}\right),$$
$$s_i = \mathrm{clip}_{[0,1]}\left[(0.30 + 0.70\beta_i)\eta_i^{\mathrm{v}}\eta_i^{\mathrm{d}}\right].$$

Thus c_i is one component of the candidate-selection hardness score s_i. The published candidate-only profile uses a uniform prior. Non-candidate fine groups and every non-tool audit column are forced to unit weight. Candidate-one fine groups use $T = 0.65$, $\delta_0 = 0.20$, $\rho = 0.80$, and the configured cap W.

The sidecar additionally records count, tool, and argument columns for audit. The executed trainer loads only the LM-compatible and auxiliary-compatible columns. In the candidate-only profile, the tool-derived weight is copied into the LM-compatible column and the auxiliary-compatible column is one. Because the count, choice, and margin losses consume the same LM-compatible row weight, equation 2 is the implementation-faithful objective.

B.5 Negative selection and sparse scoring

The hard-negative mining score is separate from the candidate-selection hardness score s_i. For target t_i, alternative $t' \in \mathcal{A}_i$, request-token set X_i, and tool-spec token set $\Phi(t')$, the mining score is

$$H_i(t') = 0.65\, R(\mathrm{name}(t_i), \mathrm{name}(t')) + 0.35\, J(X_i, \Phi(t')).$$

The candidate construction retains the target and one highest-scoring sibling negative. Both must have nonempty scored tool-name spans. The candidate branch is enabled only for one-call rows; multi-call rows receive target label -100.

Each candidate continuation is forwarded through the base model, but the FP32 log-probability calculation selects the masked tool-name positions before cross-entropy. This is algebraically identical to gathering the same positions from a dense FP32 log-softmax and avoids allocating a sequence-by-vocabulary FP32 tensor.

B.6 Candidate-focused method specification

In the candidate-focused specification, the full-target, count, choice, and margin coefficients are 1.0, 0.20, 0.30, and 0.10; the choice margin is 0.20. Protocol, argument, tool-name, and termination token weights are 1.5, 2.0, 3.0, and 4.0. All generic PGDML heads have coefficient zero. The prompt-pooled count head has nine outputs and uses multiclass negative log-likelihood; inference uses greedy generation rather than the head prediction.

Table 4: Candidate-focused specification. The executed role-stratified study uses the settings in table 5.

Setting	Value
Data / optimizer updates	79,568 rows / 4,973 configured updates
Per-device batch × accumulation	2×8 (effective 16)
Learning rate / schedule	5×10^{-6} / cosine, 3% warmup
Adapter	rank 64, LoRA alpha 128, dropout 0.05, rank-stabilized
Precision / memory	no QLoRA; bf16 compute; gradient checkpointing
Sequence / target limit	8,192 / 2,048 tokens
Candidate comparison	target plus top-ranked sibling negative; prompt budget 6,144
Weight construction	candidate-one fine groups; $T = 0.65$, floor 0.20, ESS 0.80
Selection threshold / executed cap	2 / 3
Generation	greedy; temperature 0; no count-based repair

The method selector uses the stricter threshold 2 when screening candidate configurations. The persisted sidecar supplied to training uses the executed cap $W = 3$, which is the quantity in equations (6) and (9). The two values therefore govern configuration selection and executed weights, respectively.

B.7 COMPLETED FOUR-LOSS STUDY

The six PG-U/PG-S configurations restart the final training stage from their original method-specific Stage-3 adapters. They complete 4,973 updates over all 79,568 rows with token, count, choice, and margin supervision. The remaining eighteen configurations keep their original two-loss adapters. All twenty-four are re-evaluated under the same inference protocol; no earlier prediction or score is substituted into the current matrix.

The choice branch keeps the target and at most one deterministically selected sibling. Both losses require exactly one reference call, a target present among the supplied tools, and nonempty scored tool-name spans for both target and sibling after the next-token shift. If any condition fails, both losses are zero. The margin threshold is 0.2. Losses are averaged with the same dataset-normalized row weights, without renormalizing solely over active pairs.

The new token objective weights protocol tokens by 1.5, argument tokens by 2, and tool-name tokens by 3. The first supervised assistant target token and actual native assistant EOS/EOT receive weight 8. This boundary-weighting change preserves serialization; it does not enable the auxiliary stop or repair heads. Those heads, together with plan, state, distribution, verifier-related, and reference-KL losses, retain coefficient zero.

Table 5: Executed training and evaluation settings. The new PG recipe has four active losses; the other eighteen adapters retain the earlier two-loss recipe.

Setting	Executed value
Backbones	Qwen3-4B, Qwen3-8B, Llama-3.2-3B-Instruct
Final-stage data / updates	79,568 rows / 4,973 updates, one epoch
Effective batch / seed	16 / seed 42; Qwen3-8B uses two ranks
Optimizer schedule	Learning rate 5×10^{-6}, cosine, 3% warmup
Adapter	Rank 64, alpha 128, dropout 0.05, rank-stabilized LoRA
Precision / sequence limits	BF16, no QLoRA; 8,192 context / 2,048 target tokens
New PG loss coefficients	Token 1, count 0.2, choice 0.3, margin 0.1
Control loss coefficients	Token 1, count 0.2; choice and margin disabled
New candidate budget	Two tools; maximum prompt budget 8,192 tokens
New token weighting	Protocol 1.5, names 3, arguments 2; first target/native EOS 8
Weight scope / controls	Five strata; $T = 0.7$, floor 0.2, ESS ratio 0.8, cap 3
Common evaluation	Greedy BF16, native stopping, at most 512 generated tokens

SCMD groups only by protocol stratum. For zero-call and candidate-one rows, difficulty is $\log(1 + \max\{a_i - 1, 0\})/\log 17$. For multi-call rows it combines call count and distinct-tool count with coefficients 0.6 and 0.4, clipped at eight and divided by eight; simple-one difficulty is 0.25. The score is 0.5 plus 0.5 times difficulty, clipped to $[0, 1]$. These structural weights are distinct from schema-text confusability and from the similarity score used to mine sibling negatives. Every stratum uses contraction 0.2; the minimum ESS ratio is 0.99034 and maximum normalized weight is 1.53271, so ESS and cap do not bind in this study.

PG-U and PG-S have different parent adapters on each backbone. Candidate-loss activation and boundary-token weighting also change together in the new recipe. Consequently the matrix compares complete configurations; it does not identify a same-parent SCMD effect or isolate a single loss term. The finer candidate-only weighting construction and its theoretical guarantees remain distinct from this coarser empirical intervention.

B.8 PUBLIC EVALUATION AND DENOMINATOR CONVENTIONS

The completed evaluation has 120 category inference tasks and 24 CPU-only aggregations of the same predictions, totaling 29,760 responses. All category IDs and aggregation dependencies are checked before inclusion. The fixed BFCL subset is evaluated with its AST checker and training-aligned full-call serialization. No argument repair, post-generation call truncation, or count-head decoding is applied. Native assistant stop handling is matched across all twenty-four adapters.

The parser partitions each reference population into exact count, over-call, under-call, and invalid parse. Strict zero-call recall requires a valid parse with no calls; it is not interchangeable with the BFCL Irrelevance score. All rates retain the complete reference denominator. The study uses public diagnostic data, not a full BFCL leaderboard submission, and its records explicitly mark diagnostic eligibility. Neither frozen-dev106 nor a private test split contributes to these results.

C Complete Results and Protocol Diagnostics

All tables use the completed 14 September evaluation of the same 24 adapters shown in table 1. Six are four-loss PGML reruns and eighteen are unchanged controls, all evaluated anew. The five category denominators are 400, 200, 200, 200, and 240. Values are computed from integer correct counts; no earlier partial snapshot or prediction is mixed into this matrix.

C.1 Unpooled category results

Table 6 separates Parallel and Parallel Multiple. BFCL Irrelevance and strict zero-call recall use different checking conventions: Qwen3-4B PG-S has eight Irrelevance successes but five strictly correct zero-call responses; Llama PG-S has nine and two, respectively, each out of 240. This difference reflects checking conventions, not a change of population.

Table 6: BFCL category AST accuracy for all 24 configurations (%). Multiple is single-call candidate selection. Best and second distinct values within each backbone are bold and underlined.

Method	Loss	All ↑	Simple ↑	Multiple ↑	Parallel ↑	Par.-mult. ↑	Irrel. ↑
Qwen3-4B							
LM-U	2	**29.8**	**0.5**	**1.5**	88.0	**87.0**	5.8
LM-S	2	28.4	0.0	0.0	86.0	86.0	3.3
PG-U	4	29.0	0.0	0.0	84.5	83.5	**9.6**
PG-S	4	28.3	0.0	0.0	85.5	86.0	3.3
ToolACE*	2	29.4	0.0	0.0	88.5	86.0	6.7
LESS*	2	28.6	0.0	0.0	**89.0**	85.0	2.9
NICE*	2	28.4	0.0	0.0	86.5	84.5	4.2
xLAM*	2	28.9	**0.5**	0.0	87.5	84.0	5.4
Qwen3-8B							
LM-U	2	**74.9**	**82.8**	**82.0**	92.5	89.5	29.2
LM-S	2	53.2	39.0	36.5	92.0	89.5	28.3
PG-U	4	63.0	72.0	65.5	90.0	88.0	2.5
PG-S	4	31.4	0.8	0.5	90.0	87.0	12.9
ToolACE*	2	46.1	38.3	4.0	90.0	89.0	22.1
LESS*	2	41.4	12.5	1.5	89.0	88.0	**44.2**
NICE*	2	69.5	74.8	63.5	**93.0**	**91.0**	28.3
xLAM*	2	31.1	0.3	0.0	91.5	89.5	9.6
Llama-3.2-3B							
LM-U	2	24.4	0.0	0.0	72.5	**76.5**	1.7
LM-S	2	24.4	0.3	0.0	74.5	73.0	2.5
PG-U	4	24.6	0.5	0.0	74.5	75.5	1.3
PG-S	4	25.0	0.3	0.0	75.0	75.0	3.8
ToolACE*	2	**29.2**	**6.0**	**10.0**	**79.0**	74.5	**4.6**
LESS*	2	24.1	0.0	1.0	72.0	74.5	1.7
NICE*	2	24.3	0.0	0.0	73.5	74.5	2.1
xLAM*	2	24.4	0.3	0.5	74.0	74.0	2.1

C.2 Cardinality and tool-name diagnostics

Table 7 retains all valid and invalid outputs. Count_0, Count_1, and $\text{Count}_{\geq 2}$ use 240, 600, and 400 references, respectively. Exact count, over-call, under-call, and invalid parse partition the population. Names checks tool identities and multiplicities, not arguments or call order.

Table 7: Protocol diagnostics (%). Count_0 is strict zero-call recall; Names is exact tool-name multiset accuracy. Every rate retains its full reference denominator.

Method	Loss	Parse ↑	Count ↑	Count_0 ↑	Count_1 ↑	$\text{Count}_{\geq 2}$ ↑	Over ↓	Under ↓	Names ↑
Qwen3-4B									
LM-U	2	99.8	**32.8**	5.8	**0.8**	97.0	66.0	1.0	**32.7**
LM-S	2	**99.9**	31.6	2.9	0.0	96.3	67.3	1.0	31.6
PG-U	4	99.7	32.7	**9.2**	0.0	96.0	**65.6**	1.4	**32.7**
PG-S	4	99.6	31.5	2.1	0.0	96.5	66.9	1.1	31.5
ToolACE*	2	99.8	32.4	6.3	0.0	96.8	66.6	0.7	32.4
LESS*	2	99.8	31.9	2.5	0.0	**97.3**	67.2	0.8	31.8
NICE*	2	99.7	31.7	4.2	0.0	95.8	67.1	0.9	31.5
xLAM*	2	99.6	32.3	4.6	0.3	96.8	66.7	**0.6**	32.3
Qwen3-8B									
LM-U	2	99.6	**77.8**	28.8	**84.5**	97.3	**21.0**	0.7	**77.8**
LM-S	2	99.7	55.8	27.5	39.5	97.3	43.2	0.6	55.8
PG-U	4	99.4	66.4	1.3	72.8	95.8	31.9	1.1	66.4
PG-S	4	99.2	33.1	11.7	0.7	94.8	64.8	1.2	33.1
ToolACE*	2	99.5	49.1	21.3	28.2	97.3	49.8	0.6	49.1
LESS*	2	99.7	44.4	**43.8**	9.3	97.3	54.5	0.8	44.4
NICE*	2	**99.8**	72.6	27.9	73.3	**98.3**	26.8	**0.5**	72.6
xLAM*	2	99.4	33.0	8.8	0.3	96.5	65.5	0.9	33.0
Llama-3.2-3B									
LM-U	2	**99.4**	29.8	0.0	0.0	92.5	67.7	1.9	29.7
LM-S	2	99.2	30.0	0.0	0.2	92.8	67.3	1.9	29.5
PG-U	4	99.2	30.2	0.0	0.5	93.0	66.9	2.0	29.8
PG-S	4	98.9	30.5	**0.8**	0.2	**93.8**	66.8	**1.6**	30.2
ToolACE*	2	97.9	**33.5**	0.0	**7.8**	92.3	**62.5**	1.9	**33.3**
LESS*	2	99.2	30.0	0.0	0.5	92.3	67.1	2.1	29.7
NICE*	2	98.9	30.1	0.0	0.3	92.8	66.8	2.0	29.8
xLAM*	2	99.3	30.1	0.0	0.5	92.5	67.3	1.9	29.7

Figure 4 shows how correct multi-call cardinality coexists with excessive calls on single-call and abstention references. The Llama PG-S multi-call count advantage concerns 375 of 400 cases, whereas its strict zero-call success is only two of 240. The separate denominators prevent the larger multi-call success rate from obscuring weak abstention.

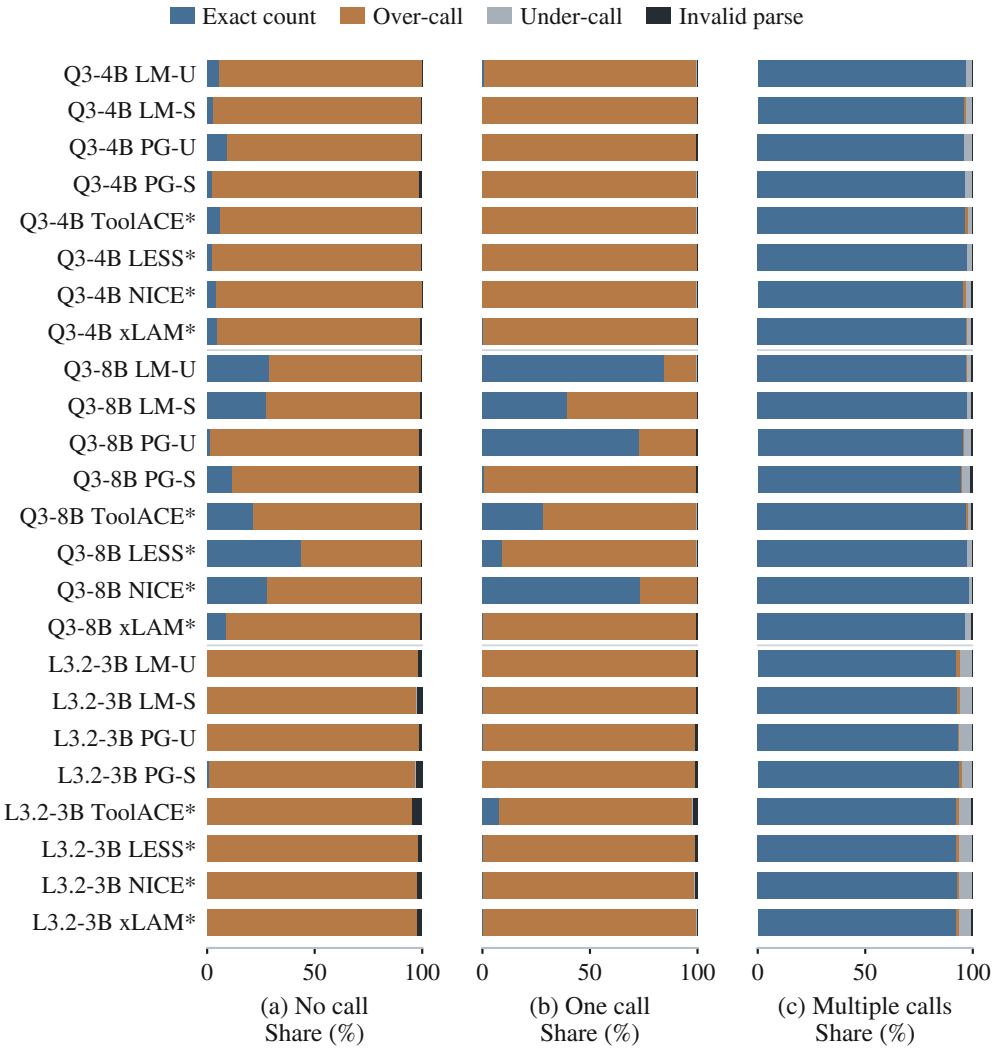

Figure 4: **How does reference call count change the error pattern?** Each bar partitions a complete reference-count stratum into exact count, over-call, under-call, and invalid parse. All 24 configurations use a common 0–100% scale; no metric conditions on successful parsing.

C.3 Scope of the comparisons

The matrix preserves inherited training-lineage differences. Six new PG recipes jointly enable candidate losses and strengthen boundary-token supervision; eighteen controls keep their original adapters. The comparisons do not isolate those changes individually or supply same-parent weighting toggles. They also do not remove count supervision or individual floor/ESS/cap constraints. All training uses seed 42, so the tables describe fixed-checkpoint outcomes rather than training-seed variance or statistical significance.

D Result Provenance and Reproduction

The numerical evidence is the complete evaluation backed up on 14 September 2026 at 13:26. It includes all 24 configurations, 120 BFCL category evaluations, and 24 protocol aggregations of the same predictions. Earlier snapshots contained 128, 137, or 142 of the 144 tasks and are excluded. The final backup contains 954 verified payload files and 29,760 prediction rows, with no missing tasks or duplicate category IDs.

The six new PG adapters are matched to completed 4,973-update training records; eighteen control adapters are matched to their retained identities. All predictions, scores, and protocol aggregates are bound to the common evaluation plan. The transfer includes results and their small provenance records, not model checkpoints or optimizer states. Integer category counts and protocol reports independently reproduce every table and plotted quantity.

The publication tables and plots are regenerated from the current merged summary using the accompanying data-to-figure script. Historical result files remain outside the manuscript's inclusion path and supply no current score. The archived plan, execution metadata, and adapter checks distinguish the six new four-loss trainings from the eighteen matched-protocol re-evaluations.

D.1 METHOD-SOURCE SCOPE

The candidate-focused method in section 3 is specified by the four-loss objective and fine-group schema-confusability construction; the completed numerical results use the role-stratified settings in appendix B.7. The current study executes the four-loss objective on six PG configurations, but does not directly test the candidate-only confusability weighting construction.

The preserved finite-core formal checks concern the mathematical construction, independently of the Protocol-Lite evaluation.