CLASS SEMANTICS ARE RESIDUAL SUPERVISION: BREADTH FIRST, DEPTH ON DEMAND

Anonymous authors
Paper under double-blind review

ABSTRACT

Many language-model applications repeatedly make decisions over a fixed output space rather than generate open-ended text, such as intent classification, request routing, and relation classification. These systems often carry an explicit semantic specification of that decision space alongside each input, including class names, definitions, and rules that distinguish neighboring labels. This raises a basic question: how much of that specification must remain explicit once the model has learned part of the decision space? We show that class semantics behave as *residual supervision*: their value is determined by the boundaries that the model and labeled examples have not yet resolved. As supervision and capability resolve more of these boundaries, the value of explicit semantics decays, and what remains concentrates on the still-unresolved boundaries. On CLINC150, the gain of full descriptions over class names falls from about $+2.5$ Macro-F1 points with five examples per class to zero with one hundred. Catalog design then becomes a semantic-allocation problem: how much depth each class should carry, and where the remaining depth should be paid for. Per-class benefit rankings change across training seeds, which makes depth-first allocation unreliable at small budgets. This instability motivates a more reliable allocation order in the regimes we study: breadth first, depth on demand. Keep a 2–4 word gloss for every class, then restore full descriptions only at unresolved boundaries. The two-pass verified compact catalog uses 40% of the full-catalog tokens and recovers 84% of its gain; the hybrid uses 58% of the tokens and bounds the loss to the full catalog at 0.4 Macro-F1 points (95% one-sided, ten seeds). Token-matched truncation and retrieval fall back to names-only performance. The remaining burden need not stay in the prompt: continuing training under the deployment prompt moves most of the catalog dependence into the model weights, recovering 97% of the accuracy lost when the catalog is removed, while a tenth of that budget recovers 11%. Catalog dependence also carries a seven-point format component that does not track the description gain. Explicit semantics are a budgeted resource: measure what remains, cover the decision space before adding depth, and move stable semantics into the weights when retraining is available.

1 INTRODUCTION

Many language-model applications repeatedly make decisions over a fixed output space rather than generate open-ended text, such as intent classification, request routing, relation classification, and product categorization. These systems often carry an explicit semantic specification of that decision space alongside each input, including class names, definitions, and rules that distinguish neighboring labels (Qu et al., 2021; Mueller et al., 2022; Pauli et al., 2023; Pattnaik et al., 2025). A common design gives every class a detailed description, with the same description included for every request.

That specification does not have to remain fixed. Pretraining already supplies some distinctions, and labeled examples resolve more of them during fine-tuning (Qu et al., 2021; Min et al., 2022). As model capability increases, the same decision space can require less explicit guidance. A description is useful only for the distinctions the model has not yet learned on its own. This raises a more fundamental question: how much of that semantic specification must remain explicit once the model has learned part of the decision space?

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.

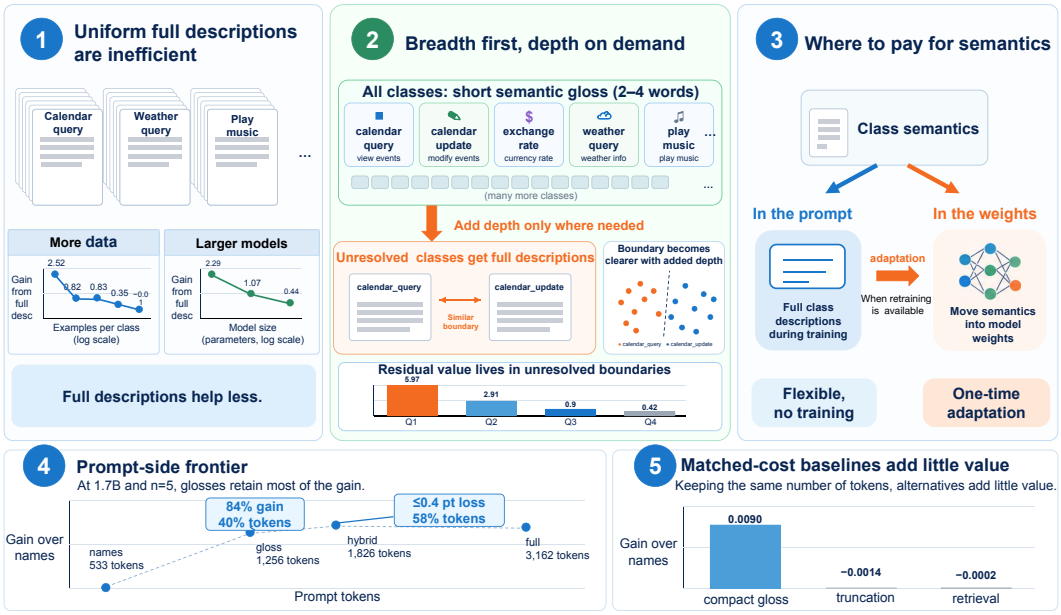

Figure 1: **Semantic supervision in a fixed decision space is residual, not permanent.** As data and model capability resolve class boundaries, the amount of explicit specification a model needs decreases and becomes localized to the boundaries that remain. Two allocation decisions follow: how to distribute the remaining semantics across the decision space, and whether to carry them in the prompt or absorb them into the weights. On CLINC150, the compact and hybrid catalogs retain most of the full-catalog accuracy with fewer prompt tokens, whereas token-matched truncation and retrieval recover little of the gain.

We find that the answer is residual. Class descriptions matter mainly for boundaries that the current model and labeled data have not yet learned to separate, so their marginal value decreases with supervision and capability, concentrates on unresolved boundaries, and can close entirely in a sufficiently resolved regime. Catalog design is then a placement question: the remaining semantic burden can sit in the prompt, in the weights, or split between them. Three measurements answer it below.

Residual demand. The gain from full descriptions over class names shrinks as labeled data or model capability increases, and it reaches zero once the training regime resolves enough boundaries on its own. On CLINC150, the full-description gain falls from about $+2.5$ Macro-F1 points with five examples per class to zero with one hundred, and at five examples from $+2.29$ points at 0.6B parameters to $+0.44$ at 4B. The decline is asynchronous: classes that are hard to separate gain much more from descriptions than easy classes, so the residual concentrates on the still-unresolved part of the label space. At low data, the least-resolved quartile gains $+5.97$ points, against $+0.42$ for the most-resolved quartile.

Coverage before depth. Concentration alone does not justify giving full text to a small subset, because such a policy must identify the right classes before deployment. Directly measured per-class response is predictive but changes sharply across training seeds; boundary statistics are more stable but less predictive. At a 25% budget, no selector separates reliably from random. A depth-first catalog therefore spends its smallest budgets on the noisiest decision in the pipeline. Those constraints fix the order in which a budget is spent. We keep a 2–4 word gloss for every class and restore full descriptions only for classes with unresolved validation boundaries. On CLINC150 at 1.7B and five examples per class, the compact catalog retains 84% of the full-catalog gain with 40% of its tokens, and the hybrid holds the full catalog's accuracy to within 0.4 points at 58% of its tokens. Coverage, content, and targeting each do distinct work.

Prompt or weights. The same residual can be paid once instead of on every request. A second training stage continues from a full-catalog checkpoint under the deployment prompt, holding the labeled data and the objective fixed. A standard-budget adaptation recovers 97% of the accuracy

lost when the catalog is removed and finishes above a model trained on names from scratch; a tenth of that budget recovers only 11%. Across eight regimes, removing the training catalog exposes a seven-point format dependence plus a semantic component that grows with the description gain, and adaptation removes both.

Together, these results give one allocation rule. Cover the label space first, add semantic depth only where unresolved boundaries remain, and, when retraining is available, move the remaining semantic burden into the weights.

2 Related Work

Label semantics in learned decision spaces. Natural-language label semantics improve data-efficient classification through entailment, semantic pre-training, and representation anchoring (Qu et al., 2021; Mueller et al., 2022; Pauli et al., 2023), and Qu et al. (2021) observe that meaningful labels matter less as supervision grows. In-context-learning studies reach a complementary conclusion from the prompt side (Min et al., 2022). The pattern now reaches past intent detection: label descriptions serve as the generation targets of a multi-label classifier (Khatuya et al., 2025), and, in fine-grained entity typing, descriptors retrieve and constrain candidate labels (Ma et al., 2026). A separate line adapts the text itself: it studies the content and quantity of intent descriptions (Hong et al., 2024), rewrites intent names from retrieved examples (Park et al., 2025), and revises descriptions from classification errors (Dev et al., 2026). These works use or optimize semantics; we ask how much of it a trained decision system still needs, and how it should be distributed across classes.

Semantic allocation in large decision spaces. Retrieval-based methods show each input only a subset of examples and labels (Milios et al., 2023; Pattnaik et al., 2025), and multi-view label representations reduce high-cardinality inference cost (Pattnaik et al., 2025). Per-request prompt compression shortens what is sent instead (Jiang et al., 2023; Mu et al., 2023), and cost-aware serving chooses model tiers (Chen et al., 2024). Both act per request on a given catalog. We treat the catalog itself as a semantic resource shared by training and serving, and study how that resource should be distributed across the decision space.

Fast decisions and deliberative reasoning. Recent work has revisited the distinction between fast, direct prediction and slower deliberative reasoning, and builds planners that trade the two off (Saha et al., 2025). That line decides when to spend more computation on a decision. Our question concerns the fast decision itself and the semantic specification it requires.

Internalizing context. Distillation transfers a larger model's behavior into a smaller one (Hinton et al., 2015), and context distillation moves information from a prompt into weights by training the model to reproduce its context-conditioned predictions without the context (Snell et al., 2022), with on-policy variants that train the student on its own generations (Agarwal et al., 2024). A related strand treats in-context adaptation and parameter updates as the fast and slow layers of one system (Tiwari et al., 2026). Existing work asks how to move contextual behavior into weights; we ask which part of a decision space still needs explicit semantics before such a transfer is worthwhile.

3 Semantic Budgeting

3.1 Supervised Classification with a Semantic Catalog

Consider a supervised classification task with K classes and training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, $y_i \in \mathcal{C} = \{c_1, \ldots, c_K\}$. In addition to labeled examples, each class c is associated with a class name n_c and a full natural-language description d_c that specifies its meaning, scope, and distinctions from neighboring classes (Hong et al., 2024; Park et al., 2025; Dev et al., 2026). We treat such class-level text as a source of *semantic supervision* (Qu et al., 2021; Mueller et al., 2022; Pauli et al., 2023). Training minimizes

$$L(\theta; \pi) = -\sum_{(x,y)\in\mathcal{D}} \log p_\theta(y \mid x, \pi), \tag{1}$$

where the sum runs over the tokens of y. A different catalog therefore changes the conditional probability of the same label and, with it, the gradient that labeled example induces. Descriptions supply no new labels; they change the condition under which the existing labels are learned (Appendix A.1).

For each class we use its name $r_c^{\text{name}} = n_c$, a 2–4 word gloss $r_c^{\text{gloss}} = (n_c, g_c)$ distilled from the description (e.g., `calendar_query`: "view existing scheduled events"), or the full text $r_c^{\text{full}} = (n_c, d_c)$. On CLINC150, full descriptions average 15 words per class and occupy about 2,600 catalog tokens. An AI model generates each gloss from its frozen description; a second AI pass verifies all entries against their sources. This two-pass catalog is frozen across models, budgets, and seeds (Appendix A).

3.2 Allocating a Semantic Budget

A *semantic allocation policy* $\pi = (r_1, \ldots, r_K)$ assigns one representation to every class, with $r_c \in \{r_c^{\text{name}}, r_c^{\text{gloss}}, r_c^{\text{full}}\}$. Given π, we fine-tune $\theta_\pi = \text{SFT}(\theta_0, \mathcal{D}; \pi)$ with the corresponding catalog and make the same catalog available at inference (Pattnaik et al., 2025). Model, data, and objective are fixed across policies; only the semantic information changes.

Let $U(\pi) = \text{MacroF1}(f_{\theta_\pi})$ denote the resulting classification utility and $C(\pi)$ the mean number of prompt tokens per request under π. Given a semantic budget B, our objective is

$$\pi_B^* = \arg \max_{\pi:\, C(\pi) \le B} U(\pi). \tag{2}$$

π shapes both fine-tuning and inference, making this a *training-time semantic allocation problem*. We compare deployable allocation orders under matched budgets; Section 4.4 tests why finite validation estimates do not reliably order a direct search.

Two uniform catalogs serve as measurement endpoints: the name-only policy π_{name} and the full policy π_{full}. Their difference $\Delta_{\text{sem}} = U(\pi_{\text{full}}) - U(\pi_{\text{name}})$ measures how much semantic supervision a given model and data regime still needs. We call it the *description gain*, the paper's measure of the residual. Every allocation policy below is scored against it. We refer to the endpoints as `P2` (names) and `P3` (names with full descriptions); Appendix H adds `P0`/`P1`.

3.3 Two Allocation Orders

Under a limited semantic budget, a natural strategy is sparse allocation (Hong et al., 2024; Park et al., 2025; Dev et al., 2026): assign full descriptions to a subset of classes while leaving the rest with names only,

$$\pi_{\text{select}}(c; S) = \begin{cases} r_c^{\text{full}}, & c \in S, \\ r_c^{\text{name}}, & c \notin S. \end{cases} \tag{3}$$

This concentrates semantic depth on a small part of the label space.

An alternative is to preserve a minimal amount of semantic information for every class before allocating additional depth selectively. We define a compact policy

$$\pi_{\text{compact}}(c) = r_c^{\text{gloss}}, \tag{4}$$

which assigns a 2–4 word gloss to every class, and a boundary-aware hybrid policy

$$\pi_{\text{hybrid}}(c; S) = \begin{cases} r_c^{\text{full}}, & c \in S, \\ r_c^{\text{gloss}}, & c \notin S, \end{cases} \tag{5}$$

where S contains classes whose learned boundaries remain unresolved according to validation behavior.

We refer to these two allocation orders as *depth-first* and *breadth-first*. Their relative value depends on the residual's size, its location, and how reliably that location can be identified before deployment.

Two places to pay for the residual. Instead of bearing the residual on every request, a system can pay for it once. Given a model trained under π and a cheaper deployment prompt π', an *adaptation* step continues training on the same labeled data under π', producing $\theta_{\pi \to \pi'}$; serving then costs $C(\pi')$ per request instead of $C(\pi)$.

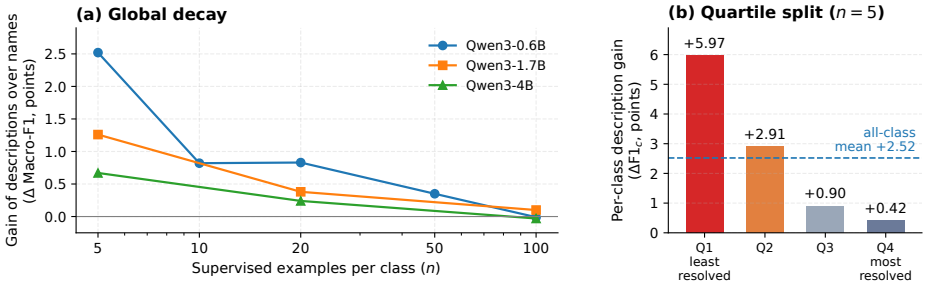

Figure 2: **Semantic demand shrinks globally and unevenly.** (**a**) Full-description gain across data and Qwen3 scales on CLINC150 (three-seed pilot grid; seven-seed anchors in text). (**b**) At n=5 and 0.6B, gain by names-only validation-recall quartile; dashed line: all-class mean (+2.52 points).

3.4 Datasets, Models, and Protocol

Our testbed is CLINC150 (Larson et al., 2019) at $n \in \{5, 10, 20, 50, 100\}$ examples per class, replicated on Banking77 (Casanueva et al., 2020) ($n \in \{5, 20\}$) and Wiki80 relation classification (Han et al., 2019). CLINC150 and Banking77 descriptions were written from public label semantics before test evaluation; Wiki80's official catalog is an independently sourced contrast. We fine-tune Qwen3 at 0.6B/1.7B/4B (Qwen Team, 2025) with one full-parameter SFT pipeline, replicating the scale axis on Gemma-3, and fix all training settings across conditions. Aggregate results use three seeds and paired bootstrap intervals; the frontier and adaptation experiments use seven seeds at 1.7B (Appendix A).

4 The Residual: Size, Location, and Ranking Instability

Uniform catalogs are justified only if all class boundaries continue to need comparable semantic detail. We test that assumption at two levels: first by varying the data and model capability available to the classifier, then by locating the remaining description gain within the label space.

4.1 Residual Demand Shrinks with Data and Capability

The marginal value of a class description falls as the training regime resolves more boundaries on its own. Both labeled data and model capability produce the same substitution. On CLINC150 with Qwen3-0.6B, the Macro-F1 gain of `P3` over `P2` falls from +2.52 points at n=5 to +0.83 at n=20 and −0.01 at n=100 (Figure 2a). Paired bootstrap intervals exclude zero at n=5 and n=20 and include zero at n=100 (Appendix B). Moving from n=5 to n=20 raises the names-only baseline by +6.4 points, far exceeding the description gain at either budget.

Model capability produces the same substitution. At n=5, the seven-seed gain falls from +2.29 points at 0.6B to +1.07 at 1.7B and +0.44 at 4B, removing 18.3%, 11.3%, and 7.7% of the respective names-only errors. At n=100, the gain is statistically zero at all three scales. Banking77 falls from +2.78 at n=5 to +1.05 at n=20. Gemma-3 reproduces the model trend (+3.29 at 1B to +1.26 at 4B); the cutoff is regime-dependent (Appendix B).

The marginal value of a description is set by how many boundaries the model and the labeled examples have already resolved, so it is neither permanent nor uniform. The aggregate curve does not yet say how to allocate a limited catalog; that requires locating the residual within the label space.

4.2 Description Gains Concentrate on Hard-to-Separate Classes

If class descriptions provide residual supervision, they should help most where the model still struggles to separate classes. The gains are highly uneven, concentrating on classes with low validation recall and strong pairwise confusion. At n=5 on the 0.6B pilot, 89 of 150 CLINC150 classes benefit from full descriptions on average and in at least two of three seeds. Grouped by names-only validation recall, the least-resolved quartile gains +5.97 F1 points, against +0.42 for the most-resolved quartile (Figure 2b).

For each class, we measure the benefit as $\Delta F1_c = F1_c(\texttt{P3}) - F1_c(\texttt{P2})$. Classes more strongly confused with a neighbor tend to gain more from descriptions: our pairwise confusion measure correlates with $\Delta F1_c$ at $+0.54$, and at $+0.35$ after controlling for recall. The concentration persists across model families and scales, and in the most-resolved classes it can reverse: on Gemma-3-1B, full descriptions reduce F1 by as much as four points. Using validation recall rather than test recall recovers 73–88% of this concentration (Appendix P).

4.3 Boundary Need Is Not the Same as Useful Content

Boundary need alone does not determine whether a description helps: the description must contain the right distinction. In three high-confusability 50-class subsets, full descriptions reduce errors among the six most-confused class pairs by 30–35% (Appendix G). Across 12 conditions, classes gain more when their descriptions actually reduce confusion with neighbors, and this relationship is stronger than the one between gain and pre-description difficulty (Appendix F). On Wiki80, official definitions change from $+0.0128$ at n=5 to -0.0134 at n=20, while a frozen boundary-aware rewrite beats them in all three seeds at n=20 by $+0.021$ (Appendix I).

4.4 Depth-First Allocation Requires an Unstable Ranking

Sparse allocation turns the catalog problem into a pre-deployment ranking problem. The direct signal, R_c (per-class validation-F1 difference between paired `P3` and `P2` checkpoints), predicts held-out benefit best (Spearman $+0.38$ at 1.7B), but its ranking reproduces across seeds at only $+0.09$. A coarse boundary statistic is less predictive ($+0.15$) but far more reproducible ($+0.63$). At a fixed 25% budget, response, difficulty, and random selection are statistically indistinguishable after retraining (Appendices C and D).

Larger estimation sets improve predictive power but do not close the stability gap (Appendices M and O). Banking77 provides the clearest separation: at n=20 the full catalog still adds $+0.0105$, yet the response ranking approaches noise and the resulting allocation recovers none of that gain (Appendix V). A response ranking measured at 0.6B and deployed at 1.7B preserves 97% of what target-side measurement recovers, reaching $+0.0069$ over `P2` at 1,256 tokens where the gloss reaches $+0.0090$ (Appendix L).

5 Breadth First, Depth on Demand

Once the residual has been measured, the next question is how to allocate it across the decision space. When the prompt that trains the model must also serve it, the residual has to be carried in that prompt, and Section 4 leaves three constraints: demand is uneven, useful content matters more than length, and the class ranking that sparse allocation needs is noisy. Breadth first, depth on demand meets all three by postponing the ranking decision. Every class receives a 2–4 word gloss, and full descriptions return only where validation boundaries remain unresolved.

We measure the complete frontier on CLINC150 at 1.7B and n=5, using seven seeds unless noted otherwise. A compact gloss for all 150 classes costs 1,256 mean prompt tokens; a depth-first catalog with full descriptions for only the 38 least-resolved classes costs 1,285, within 3%, isolating the order in which the budget is spent.

5.1 Coverage Comes Before Depth

A compact gloss for every class retains most of the full-catalog gain with a fraction of its tokens. The verified compact catalog recovers 84% of the full-catalog gain using 40% of the tokens, 0.17 points below the full catalog (Table 1). The hybrid uses 58% of the tokens and falls within 0.4 points of the full catalog (95% one-sided; Appendix N). Selective top-75 reaches a higher Macro-F1 than the full catalog (119% capture) because it omits descriptions from resolved classes where they were slightly harmful (Appendix J), but it costs 39% more tokens than the hybrid.

Table 1: **The semantic-budget frontier** (CLINC150, Qwen3-1.7B, n=5, 7 seeds). All policies share the same training protocol; only the catalog differs. Capture expresses each policy's gain as a share of the full-catalog gain; the claims rest on absolute Macro-F1 and paired tests.

catalog policy	Macro-F1	vs. P2	capture	tokens	gain / 1k tok
P2 names	0.9052	—	0%	533	—
compact gloss (all 150)	**0.9142**	**+0.0090**	**84%**	**1,256**	**7.13e-3**
compact + full text, top-38	**0.9161**	**+0.0109**	**102%**	**1,826**	5.97e-3
compact + full text, top-75	0.9129	+0.0076	72%	2,291	3.33e-3
compact + full text, *random* 38	0.9106	+0.0054	51%	1,734	3.11e-3
selective full text, top-25 (1.7B difficulty)	0.9118	+0.0065	61%	1,285	5.08e-3
selective full text, top-75 (1.7B difficulty)	**0.9179**	+0.0127	119%	2,551	4.96e-3
full catalog	0.9159	+0.0107	100%	3,162	3.37e-3

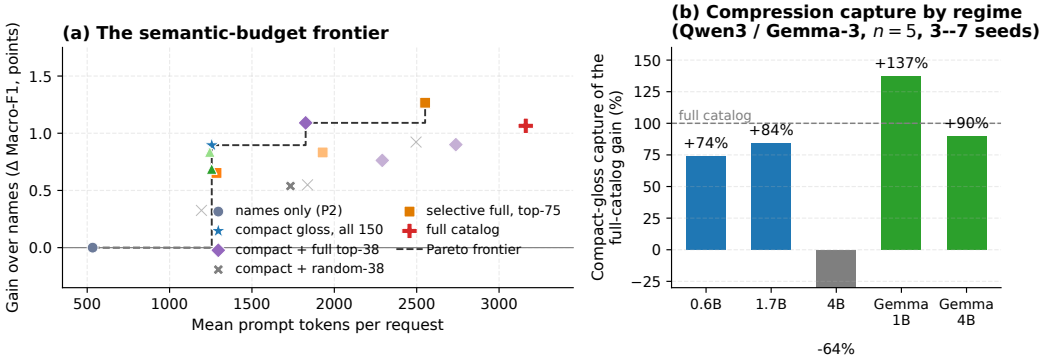

Figure 3: **The semantic-budget frontier.** (**a**) Accuracy gain over names versus mean prompt tokens at 1.7B and n=5; the dashed staircase is the Pareto frontier. The hybrid reaches the full catalog's accuracy range with 42% fewer tokens. (**b**) The compact gloss retains most of the full-catalog gain when unresolved boundaries are broad, but not at Qwen3-4B, where the residual is smaller and more localized.

5.2 Coverage, Content, and Targeting Are Distinct

Three matched controls separate the ingredients of the breadth-first advantage. At roughly the gloss's cost, verbatim truncation reaches 0.9038 and per-input retrieval 0.9050, versus 0.9052 for names and 0.9142 for glosses (Appendices T.1 and T.2). Boundary content and label-space coverage both matter. Catalog C reproduces the hybrid's frontier position, while its targeted-versus-random contrast remains unresolved ($+0.0019$, $[-0.0045, +0.0083]$; Appendix S). In the verified catalog, targeting the 38 unresolved classes reaches 0.9161 against 0.9106 for a random subset ($+0.0055$, $[+0.0010, +0.0100]$); random depth is worse than the compact gloss alone despite using 38% more tokens.

Text at a resolved boundary can pull neighboring inputs into the class: removing descriptions from the 38 most-resolved classes cuts false positives into them and raises their held-out F1, while a same-size random removal costs the removed classes accuracy (Appendix J).

5.3 Depth Follows the Residual

The required depth tracks the residual: across regimes where full descriptions add broad value, the compact gloss retains 74–137% of their gain with about 40% of their tokens. At Qwen3-4B, most boundaries are already resolved: the gloss adds nothing over names, while restoring full text for the 38 least-resolved classes recovers it (Appendix Q). Weaker models need more depth: at 0.6B, top-75 reaches 104% of the full-catalog gain, and on Banking77 top-58 reaches 102% (Appendices R and W). The ordering holds across regimes; the value of its first step does not.

6 MOVING THE RESIDUAL INTO THE WEIGHTS

Section 5 allocates the residual within a catalog used for both training and serving. A stable decision space permits another placement: train with the full catalog, then adapt under the cheaper deployment prompt. The model learns to operate without repeated descriptions. Simply deleting them from a full-catalog checkpoint loses many points (Appendix J); the second stage teaches the model to use the prompt it will see in service.

Starting from the checkpoints used above, we continue training on the same labeled examples under the *deployment* prompt, either class names or the compact gloss. The standard recipe uses lr 2×10^{-5} for at most ten epochs and selects the best epoch under that prompt. An equal-budget control continues under the rich prompt, isolating prompt adaptation from extra optimization. A gentle recipe, with one tenth of the learning rate and at most four epochs, tests whether a small update is sufficient. Appendix K gives the full protocol.

6.1 ADAPTATION RECOVERS THE RESIDUAL

A standard-budget adaptation recovers 97% of the accuracy lost when the catalog is removed and finishes above a model trained on names from scratch. On CLINC150 at 1.7B and n=5, names-prompt accuracy rises from 0.8020 before adaptation to 0.9122 after it, recovering 65% of the full catalog's advantage over names-only training and finishing 0.70 points above names-from-scratch (7/7 seeds, 95% CI $[+0.0029, +0.0110]$). The edge persists against a two-stage names-only control: $+0.0061$ ($[+0.0011, +0.0112]$, 6/7 seeds), and $+0.0110$ ($[+0.0038, +0.0182]$) when the control's second-stage budget targets the rich path's selected total epochs per seed (Appendix K). The equal-budget control reaches only 0.8221, and the rich prompt does not decline. The gentle recipe recovers 11%.

6.2 THE RESIDUAL HAS TWO COMPONENTS

Removing the catalog from a rich-trained checkpoint costs points in proportion to the description gain, plus a positive intercept that persists even where descriptions added no accuracy during training. Across eight regimes, the loss is $6.98 + 2.46\,\Delta_{\text{sem}}$ points (r=0.80, R^2=0.64; Figure 4). The intercept is a format effect: at CLINC150 0.6B and n=100, descriptions add no accuracy during matched training, yet removing them still costs 6.8 points. Prompt format outlives the information it once carried.

Adaptation removes both components in every tested cell. Its advantage over names-from-scratch concentrates in the three high-residual regimes ($+0.0070$ to $+0.0123$) and is indistinguishable from zero in the other five. At the class level it concentrates in the least-resolved quartile (Appendix K): adaptation preserves the residual rather than creating new signal.

6.3 SERVING COST

On CLINC150, the names-adapted model reaches an accuracy range comparable to the compact and hybrid prompts while using 533 tokens instead of 1,256 or 1,826. On Banking77, adaptation to the compact gloss reaches 0.7877, above both the full catalog (0.7835) and the best prompt-side frontier point (0.7841), with 42% of the full-catalog tokens. The measured one-time cost is 27 minutes on one 80 GB accelerator at 1.7B; names-only serving then saves 2,629 tokens per request.

7 DISCUSSION

Semantic specification in a fixed decision space is a learned resource. What a system must keep explicit depends on what its model and data have already resolved. One measurement and one deployment constraint determine where that resource should live: the description gain and whether the model can be retrained. Together they sort the decision space into three cases.

No residual demand. When the full-versus-names gap has closed, semantic depth buys no accuracy and names are enough. The checkpoints in hand still need care. At n=100 on CLINC150,

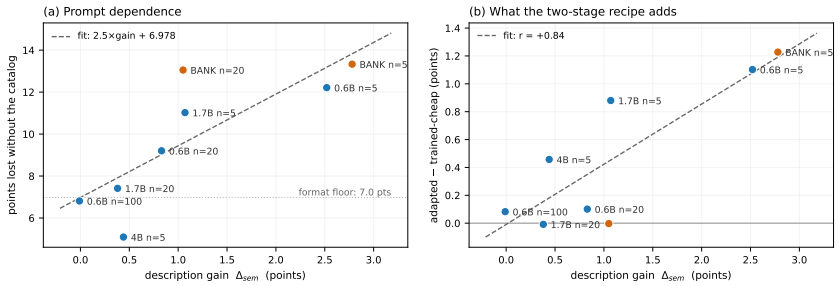

Figure 4: **Where the catalog's value goes.** Each point is one cell (dataset, scale, data budget; 3 seeds, 7 for CLINC150 1.7B n=5). (**a**) Points lost by the rich-trained model when its catalog is removed, against the description gain the paper measures for the same cell; the fit has a positive intercept. (**b**) What the standard-budget adaptation buys over training on names from scratch, against the same gain.

matched names-only and full-catalog training reach the same accuracy, yet deleting the catalog from the full-trained checkpoint loses 6.8 points. A closed description gain is therefore a reason to train and serve with names in the next model, not a license to change an existing model's prompt without adaptation.

Residual demand in a shared catalog. When one catalog must serve both training and inference, the residual stays in the prompt and the breadth-first frontier allocates it. The compact gloss is the first increment because it requires no per-class ranking; full text is added only after validation evidence identifies unresolved boundaries. The hybrid selects the 38 least-resolved classes from each seed's names-only validation recall, which test labels never enter. That statistic identifies difficult boundaries rather than measuring a description's direct effect, so the depth step needs a held-out set representative of the boundaries the system will face.

Residual demand with retraining available. When the model can be adapted, the residual can be paid once. In the high-residual regimes tested here, rich-then-cheap adaptation finishes 0.7–1.2 points above training directly with the cheap prompt, and it removes the format dependence as well, so serving runs on names with no recurring semantic cost.

Tokens are a portable cost axis. Compact and hybrid catalogs cut recurring context to 40% and 58% of the full catalog, and adaptation reduces it to class names after one fine-tune. We count prompt tokens because they transfer across serving stacks; what a token actually costs depends on prefix caching, which is deployment-specific (Appendices L and K).

Scope and limitations. The evidence covers full-parameter SFT with 77–150 labels and 0.6B–4B backbones; larger taxonomies, models, and production serving remain unmeasured. Independent gloss-generation recipes recover 36–70% of the full-catalog gain, versus 84% for the two-pass AI-verified catalog (Appendix S). Substitution results use three seeds while the frontier and adaptation use seven; the hybrid reads a signal estimated from a finite validation split (Appendix M). The seven-point format component outlives the catalog's information, and where it comes from is an open question.

8 Conclusion

Class semantics are residual supervision: their value falls as data and model capability resolve boundaries and concentrates where ambiguity remains. A shared catalog should cover the decision space before adding depth; when adaptation is available, stable semantics can move into the weights. Larger decision spaces may exceed even compact coverage, calling for retrieval, selective depth, and weight transfer within one budget.

AI USE STATEMENT

Generative AI tools were used to assist with code writing, generate early visual references for figure design, organize relevant literature, and polish the manuscript. The early visual references were used only during design iteration and were not treated as scientific evidence. The authors reviewed and revised all AI-assisted content, verified the final code, figures, citations, and empirical claims, and take full responsibility for the final manuscript and its supporting artifacts.

ETHICS STATEMENT

This work uses previously released research datasets and collects no new data from human participants or new personally identifying information. Dataset use and artifact release follow the applicable source licenses. Released materials will exclude credentials, private infrastructure paths, and data that cannot be redistributed.

REPRODUCIBILITY STATEMENT

Appendix A documents data splits, catalog provenance and hashes, and the training protocol. The remaining appendices report per-condition results, probe definitions, controlled interventions, and per-seed statistics. The supplementary package provides deterministic data-preparation and job-generation scripts, frozen configurations and manifests, analysis code, and per-seed metrics. Test labels are excluded from description selection.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations (ICLR)*, 2024.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI (NLP4ConvAI)*, pp. 38–45, 2020.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=cSimKw5p6R.

Hamvir Dev, Shivam Ratnakant Mhaskar, Sasanka Vutla, and Anup Pattnaik. Beyond instruction optimization: Multi-agent error-driven class description refinement for llm-based classification. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 2031–2049, 2026.

Xu Han, Tianyu Gao, Yuan Yao, Deming Ye, Zhiyuan Liu, and Maosong Sun. OpenNRE: An open and extensible toolkit for neural relation extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pp. 169–174, 2019. doi: 10.18653/v1/D19-3029. URL https://aclanthology.org/D19-3029/.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

Taesuk Hong, Youbin Ahn, Dongkyu Lee, Joongbo Shin, Seungpil Won, Janghoon Han, Stanley Jungkyu Choi, and Jungyun Seo. Exploring the use of natural language descriptions of intents for large language models in zero-shot intent classification. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pp. 458–465, 2024.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Subhendu Khatuya, Shashwat Naidu, Saptarshi Ghosh, Pawan Goyal, and Niloy Ganguly. Label-semantics aware generative approach for domain-agnostic multilabel classification. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 22286–22298, 2025. doi: 10.18653/v1/2025.findings-acl.1145. URL https://aclanthology.org/2025.findings-acl.1145/.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1311–1316, 2019. doi: 10.18653/v1/D19-1131. URL https://aclanthology.org/D19-1131/.

Yingying Ma, Hongliang Dai, Xia Zhang, and Piji Li. Leveraging label semantics and entity description generation for LLM-based fine-grained entity typing. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 31655–31667, 2026. doi: 10.18653/v1/2026.findings-acl.1584. URL https://aclanthology.org/2026.findings-acl.1584/.

Aristides Milios, Siva Reddy, and Dzmitry Bahdanau. In-context learning for text classification with many labels. In *Proceedings of the 1st GenBench Workshop on (Benchmarking) Generalisation in NLP*, pp. 173–184, 2023. doi: 10.18653/v1/2023.genbench-1.14.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. Learning to compress prompts with gist tokens. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Aaron Mueller, Jason Krone, Salvatore Romeo, Saab Mansour, Elman Mansimov, Yi Zhang, and Dan Roth. Label semantic aware pre-training for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8318–8334, 2022.

Gyutae Park, Ingeol Baek, Byeongjeong Kim, Joongbo Shin, and Hwanhee Lee. Dynamic label name refinement for few-shot dialogue intent classification. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 41–52, 2025.

Anup Pattnaik, Sasanka Vutla, Hamvir Dev, Jeevesh Nandan, and Cijo George. Scalable and cost effective high-cardinality classification with LLMs via multi-view label representations and retrieval augmentation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 1955–1969, 2025. doi: 10.18653/v1/2025.emnlp-industry.137.

Amalie Pauli, Leon Derczynski, and Ira Assent. Anchoring fine-tuning of sentence transformer with semantic label information for efficient truly few-shot classification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 11254–11264, 2023.

Jin Qu, Kazuma Hashimoto, Wenhao Liu, Caiming Xiong, and Yingbo Zhou. Few-shot intent classification by gauging entailment relationship between utterance and semantic label. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI (NLP4ConvAI)*, pp. 8–15, 2021.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Swarnadeep Saha, Archiki Prasad, Justin Chih-Yao Chen, Peter Hase, Elias Stengel-Eskin, and Mohit Bansal. System-1.x: Learning to balance fast and slow planning with language models. In *International Conference on Learning Representations (ICLR)*, 2025.

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context. *arXiv preprint arXiv:2209.15189*, 2022.

Rishabh Tiwari, Kusha Sareen, Lakshya A Agrawal, Joseph E. Gonzalez, Matei Zaharia, Kurt Keutzer, Inderjit S Dhillon, Rishabh Agarwal, and Devvrit Khatri. Learning, fast and slow: Towards llms that adapt continually. *arXiv preprint arXiv:2605.12484*, 2026.

Appendix roadmap. The appendix verifies four links in the main argument. Appendix A records the protocol. Appendices B, F, G, I, and P establish the size of the residual and its location at unresolved boundaries. Appendices C, D, M, O, and V test the ranking instability that makes small-budget depth-first allocation unreliable. Appendices N, Q, R, S, T, and W verify the breadth-first frontier across tolerances, regimes, recipes, and matched-cost alternatives. Appendix K reports the weight-side transfer and its cost; the remaining sections provide transfer, per-seed, and prompt-only checks.

A Reproducibility Details

Data. CLINC150 source file SHA-256 `36923c37...d56e0`; balanced subsets are generated per (budget, seed) with manifests recording class counts. Banking77 slices include manifests and hashes. Validation (3,000) and test (4,500) splits are identical across all budgets.

Descriptions. The frozen CLINC150 description file (`descriptions.FROZEN.json`, SHA-256 `cf8cac9cde8cbbc430cda982a4accec19a182824c6dacee6e1b83f20376bfe8c`) covers all 150 classes, was written from public label semantics without access to test predictions, and explicitly documents boundaries for high-similarity class pairs. Banking77 uses a description file written under the same protocol.

Compact-gloss protocol. We made one AI compression pass over the frozen CLINC150 descriptions, followed by a second AI pass that verified all 150 entries against their frozen source descriptions. The verification instruction, per-class verdicts, and cue audit are archived as `descriptions.UNIFORM_SHORT.verify.json` (147 pass, 3 pass with scope notes, 0 fail; 44 classes exempt from the anti-collision cue; generator script `scripts/verify_gloss_catalog.py`). The instruction was: express the affirmative intent in two to four words; retain an object, state, or operation that separates the class from its closest lexical neighbor when it fits; do not add information absent from the frozen description. Construction used no training, validation, or test examples, no model outputs or error analysis, and no class-benefit ranking. The resulting file was frozen before the compact, hybrid, truncation, and cross-regime runs and reused without model- or seed-specific edits (`descriptions.UNIFORM_SHORT.json`, SHA-256 `826621f272e4eae07be9756470d4dea22eab49f523f30f04b4e9813a0a808aac`). Examples are `calendar`: "view existing scheduled events," `calendar_update`: "modify existing calendar events," `bill_balance`: "bill amount currently owed," and `bill_due`: "payment due date." The token audit reports 2–4 tokens per entry (3.9 mean under the Qwen3 tokenizer).

Training. Full-parameter SFT, lr 2×10^{-5}, batch $2\times$grad-accum 8, max length 6,144, $\leq$10 epochs, patience 3, best checkpoint by validation Macro-F1 per epoch. Loss on the completion token only.

A.1 Completion-Only Loss and Conditioning Semantics

Fine-tuning minimizes the completion-only objective

$$L(\theta;\pi) = -\sum_{(x,y)\in\mathcal{D}} \log p_\theta(y \mid x, \pi), \tag{6}$$

where the sum runs over the tokens of y and π enters only as conditioning context. Changing π changes $L(\theta;\pi)$ and, at any fixed θ, the gradient that the same labeled example contributes, so $\nabla_\theta L(\theta;\pi_{\text{name}})$ and $\nabla_\theta L(\theta;\pi_{\text{full}})$ are different vectors in general. This is what the term *semantic supervision* refers to: a description adds no label, but it changes the condition under which the existing label is predicted, and therefore which errors the update corrects. The quantity the paper measures is the observable consequence of that difference, $\Delta_{\text{sem}} = U(\pi_{\text{full}}) - U(\pi_{\text{name}})$, the marginal utility of

additional conditioning semantics under a given model and data regime. The paper's claims concern Δ_{sem} and how it distributes across classes rather than a decomposition of this gradient.

B AGGREGATE PRIOR-UTILITY TABLES

Table 2: **Data axis of the trade-off on CLINC150** (Qwen3-0.6B, test Macro-F1, mean $\pm$ std over 3 seeds). Δ = P3$-$P2; CIs are per-seed paired bootstrap 95%.

Examples per class	P2 (names)	P3 (names+desc.)	Δ	significance
5	0.8754 ± 0.0053	0.9006 ± 0.0062	**+0.0252**	all seeds > 0
10	0.9181 ± 0.0026	0.9264 ± 0.0039	+0.0082	2/3 seeds > 0
20	0.9397 ± 0.0022	0.9481 ± 0.0019	+0.0083	all seeds > 0
50	0.9604 ± 0.0010	0.9639 ± 0.0006	+0.0035	CIs include 0
100	0.9680 ± 0.0016	0.9679 ± 0.0017	-0.0001	CIs include 0

Table 3: **Model axis of the trade-off on CLINC150** (3-seed pilot grid). Each cell reports absolute Δ Macro-F1 and, in brackets, $\Delta/(1-\text{P2})$: the gain as a share of remaining names-only error. Section 4 reports the same n=5 anchor on the 7-seed frontier grid ($+2.29/+1.07/+0.44$ for 0.6B/1.7B/4B); the two batches share the protocol and differ only in seed set.

Model	$n = 5$	$n = 20$	$n = 100$
Qwen3-0.6B	+0.0252 [20.2%]	+0.0083 [13.8%]	-0.0001 [-0.2%]
Qwen3-1.7B	+0.0126 [13.0%]	+0.0038 [7.4%]	+0.0010 [3.6%]
Qwen3-4B	+0.0067 [11.8%]	+0.0024 [7.3%]	-0.0003 [-1.3%]

Table 4: **The substitution curve across families and data budgets** (CLINC150, `P2` vs. `P3`, Macro-F1; 7 seeds for all Qwen3 rows and 3 seeds for Gemma-3). Model scale moves the description gain down within each family. The data-side cutoff is regime-dependent: at n=100 the 1B Gemma model still converts descriptions into $+2.7$ points, while the 4B Gemma model is near-neutral at n=20 ($+0.03$) and stays near-neutral at n=100 ($+0.22$).

Family	condition	`P2`	`P3`	gain
Qwen3	0.6B, n=5	0.8751	0.8980	$+2.29$
	1.7B, n=5	0.9052	0.9159	$+1.07$
	4B, n=5	0.9431	0.9475	$+0.44$
Gemma-3	1B, n=5	0.8548	0.8878	$+3.29$
	1B, n=20	0.9008	0.9270	$+2.62$
	1B, n=100	0.9198	0.9469	$+2.71$
	4B, n=20	0.9659	0.9662	$+0.03$
	4B, n=100	0.9745	0.9767	$+0.22$

C ALLOCATION SIGNALS: ACCURACY VERSUS REPRODUCIBILITY

Table 5: **Allocation signals: predictive power vs. ranking reproducibility.** Reproducibility is the mean seed-pair Spearman of the class ranking; predictive power is the Spearman with held-out per-class gain. Measured response ranks classes best and reproduces least reliably; the boundary statistic is the reverse.

Scale	signal	reproducibility	predictive power
0.6B	response	$+0.28$	$\mathbf{+0.50}$
	difficulty	$\mathbf{+0.68}$	$+0.33$
1.7B	response	$+0.09$	$\mathbf{+0.38}$
	difficulty	$\mathbf{+0.63}$	$+0.15$
	confusion	$+0.61$	$+0.21$
cross-scale	response	$+0.16$ (0.6B→1.7B rank agreement)	
	difficulty	$+0.59$	
	confusion	$+0.58$	

D Static Boundary Selectors at a Fixed Budget

At the 1.7B/n=5 frontier, the three token-matched 25% selectors are statistically indistinguishable after retraining: response-selected 0.9121 (vs. random $+0.0036$ $[-0.0004, +0.0077]$), difficulty-selected 0.9136 ($+0.0051$ $[-0.0001, +0.0104]$), and random 0.9085, over 7 seeds, with every paired interval including zero. The comparison that decides the paper is therefore over description length (Section 5), not over which classes are selected. The same comparison at the larger depth budgets in Table 1 keeps the sign but not the significance: the target-scale difficulty ranking is $+0.0033$/$+0.0028$/$+0.0034$ above random at 25/50/75% of classes, with every interval containing zero.

Does targeting matter? A random-subset control. The hybrid policies of Table 1 share one breadth and differ only in which K classes receive full text; on their own they cannot tell us whether the *choice* matters or whether extra text would do anywhere. We therefore trained *random-38*: the identical compact catalog with full descriptions for 38 classes drawn at random. At the same breadth and K=38, the targeted hybrid reaches 0.9161 against 0.9106 for the random subset (-0.0055, $[-0.0100, -0.0010]$, 6/7 seeds; seed 404 is the exception at $+0.0040$), and the random-depth catalog is *worse* than the compact gloss alone (-0.0037, $[-0.0068, -0.0006]$) despite 38% more tokens. Holding the described classes fixed at that random subset and varying only whether the other 112 carry glosses or names gives $+0.0021$ ($[-0.0054, +0.0097]$); the combined effect against the random-selective baseline of Table 1 is $+0.0077$ ($[+0.0032, +0.0122]$). The 38 selected classes also carry slightly longer text than a random 38 (1,826 vs. 1,734 mean prompt tokens), so the targeting advantage is not a token artifact. This is the identification the allocation claim needs: at this budget, depth pays where the boundary is unresolved and costs accuracy when spent elsewhere—the aggregate signature of the attraction effect in Appendix J. The subsets behind these runs are recorded per seed with the artifacts.

Difficulty and confusion do correlate with description benefit (Section 4.2), suggesting a cheap recipe: select classes directly from the names-only model's statistics. We test it on CLINC150 by retraining with each selection (Table 6); every selector receives the same fixed 25% class budget, and we compare a random subset, **Difficulty**, **Confusion**, and their equal-weight rank fusion **Difficulty+Confusion**.

Table 6: **Static boundary selectors under a fixed budget** (CLINC150, Qwen3-0.6B, test Macro-F1, mean over 3 seeds; parenthesized: share of the full-description gap). Every selector describes 25% of classes.

n	P2	Full P3	Random	Difficulty	Confusion	Diff.+Conf.
5	0.8754	0.9006	0.8886 (53%)	0.8913 (63%)	0.8884 (52%)	0.8799 (18%)
20	0.9397	0.9481	0.9427 (35%)	0.9415 (21%)	0.9432 (42%)	0.9399 (2%)
100	0.9680	0.9679	0.9665	0.9675	0.9687	0.9669

Static scores diagnose aggregate susceptibility but do not produce a stable top-ranked set: difficulty captures 63% of the full-description gain at n=5 but 21% at n=20, while rank fusion captures 18%

and 2%. Allocation needs evidence about the exact class–description pair because the same observed difficulty may reflect an under-sampled boundary, intrinsic ambiguity, or a distinction the catalog can directly supply. Wrong-prior interventions independently confirm this content dependence (Appendix U).

E Performance–Prior-Cost Pareto

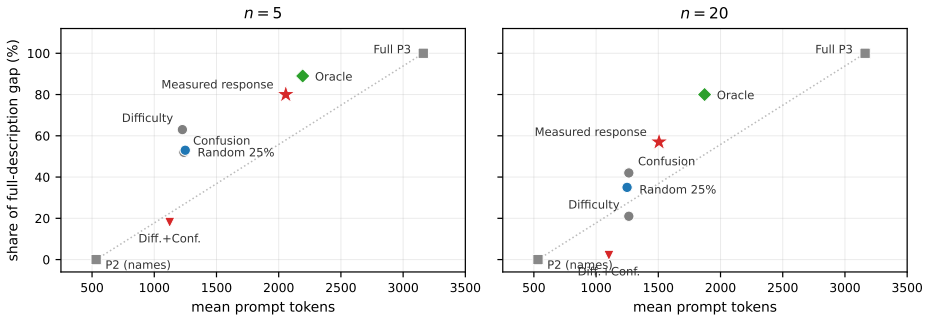

Figure 5: **Performance–prior-cost Pareto** (CLINC150, 0.6B). Share of the full-description gap recovered versus mean prompt tokens at n=5 and n=20. Unlike the fixed-25% main comparison, the response points here use the adaptive rule $R_c > 0$.

The adaptive dev-response rule obtains Macro-F1 $0.8955/0.9445/0.9701$ at n=5/20/100 (versus P2 $0.8754/0.9397/0.9680$), selecting roughly $55\%/35\%/22\%$ of classes. Its test-side oracle obtains $0.8977/0.9464/0.9703$. These results trace the shrinking useful share; the main text fixes a 25% budget for controlled selector comparisons.

F Boundary Probe: Protocol and Quadrants

For each evaluated (dataset, model, budget, seed) condition we score the same fixed validation split with both the `P2` and `P3` checkpoints, producing candidate-level log-probabilities over the full class inventory. This includes the 0.6B data grid and the completed 1.7B and 4B probes at CLINC150 n=5. Writing $s_\ell(x)$ for label score, we compute $m(x) = s_y(x) - \max_{j \neq y} s_j(x)$, $B_c = \mathbb{E}_{x \in c}[\text{softplus}(-m(x))]$, and $\Delta B_c = B_c^{P_2} - B_c^{P_3}$.

Table 7: **Boundary probes.** Seed-averaged Spearman correlations of class-level gain with initial uncertainty and measured repair; repair is positive in all 12 conditions.

Dataset / setting	n or scale	$\rho(B_c^{P_2}, \Delta F1_c)$	$\rho(\Delta B_c, \Delta F1_c)$
CLINC150 (data; 0.6B)	5/10/20/50/100	0.35/0.23/0.10/0.14/0.18	0.56/0.42/0.43/0.42/0.51
CLINC150 (scale; n=5)	1.7B/4B	0.37/0.14	0.51/0.36
Banking77 (0.6B)	5/20	0.34/0.27	0.49/0.35
Wiki80 (0.6B)	5/20/100	−0.16/0.02/−0.02	0.53/0.62/0.33

Median-split quadrants (CLINC150, n=5, seed-averaged): high-need/repaired classes gain mean $\Delta F1 = +0.0736$ (48 classes); high-need/unrepaired $+0.0014$ (27); low-need/repaired $+0.0276$ (27); low-need/unrepaired -0.0077 (48).

G Controlled Label-Space Manipulation

To test whether description gain depends on the confusability structure of the label space itself, we manipulate it while holding everything else constant: three matched subset pairs (high/low, 50 classes each) drawn from CLINC150, with the high-confusability side selected for maximal symmetric confusion while matching per-class recall distributions ($|\Delta\text{mean recall}| \leq 0.01$, pairwise

Jaccard ≤ 0.6). Budget ($n \in \{5, 20\}$), models, examples per class, and train/test prompts are identical to the main grid. Each condition trains one seed (Qwen3-0.6B, n=5 and n=20, `P2` and `P3`).

Table 8: **Controlled label-space experiment** (Δ = P3−P2 Macro-F1, seed 13). The aggregate high–low contrast is positive in 3/3 pairs at n=20 and mixed at n=5; localized confusion-pair repair is stable across both budgets.

Pair	n	Δhigh	Δlow	Δhigh − Δlow
h1/l1	5	+0.0192	+0.0259	−0.0067
h2/l2	5	+0.0442	+0.0305	+0.0137
h3/l3	5	+0.0059	+0.0355	−0.0296
h1/l1	20	+0.0120	+0.0061	+0.0058
h2/l2	20	+0.0112	+0.0003	+0.0108
h3/l3	20	+0.0126	+0.0114	+0.0011

The trained `P2` models preserve the intended ordering: the high side carries more confusion in all three pairs. Supervision compresses the designed gap from ≈ 0.045 to 0.006–0.011, making localized errors the most informative readout. On every high-confusability subset, the top-6 symmetric confusion pairs concentrate the errors, and descriptions reduce these counts from 159 to 111 at n=5 (−30%) and from 94 to 61 at n=20 (−35%). This stable pairwise repair supports the localization account in Section 4.3 without requiring a uniform aggregate high–low effect.

H Empirical Prior Ladder (Supporting Analysis)

To decompose the information sources inside the prior, we add two auxiliary conditions to `P2` and `P3`: `P1` (opaque labels `label_01,...,label_K`, replacing class names) and `P0` (no fixed class catalog; the model must generate valid class names itself). Results on CLINC150 at n=5 (Qwen3-0.6B, mean over 3 seeds):

Table 9: **Empirical prior ladder** (CLINC150, n=5). Increments are Macro-F1 gains over the previous rung; adjacent rungs jointly change semantic information, output space, and generation difficulty.

Condition	Macro-F1	over previous
`P1`: opaque labels	0.2378	—
`P0`: no fixed catalog	0.8210	+0.583
`P2`: class names	0.8754	+0.054
`P3`: names + descriptions	0.9006	+0.025

Moving from opaque labels to usable class names yields the largest gain; fixing the class space helps further; detailed descriptions add the smallest rung. All gaps collapse with data (the whole ladder spans 1.4 points at n=100). The highest-gain classes include adjacent-pair ambiguities such as `change_ai_name`/`change_user_name` and `freeze_account`/`account_blocked`; semantically distinct classes such as `book_flight` and `pin_change` gain little.

I Wiki80 as a Boundary Condition: Official vs. Boundary-Aware Descriptions

Wiki80 cleanly separates semantic availability from semantic usefulness: names-only performance is 0.478 Macro-F1 at n=5, while official Wikidata descriptions add +0.0128 and turn neutral by n=20. Two targeted interventions identify which source supplies usable information (Qwen3-0.6B, 3 seeds; the P1 ladder additionally at n=100):

(i) Prior ladder with opaque labels. We train `P1` with labels `label_01,...,label_80`. The resulting `P1`–`P2` gap measures the semantic contribution of relation names, while the `P2`–`P3` gap measures the added value of official definitions.

(ii) Boundary-aware descriptions. We rewrite all 80 descriptions as "definition + explicit boundary": each states the relation and distinguishes its nearest confusable neighbors (e.g., `country of citizenship` versus `country of origin`). Authored from Wikidata property semantics alone and frozen before evaluation (SHA-256 recorded), the catalog tests whether explicit distinctions convert unresolved boundaries into gain. The rewrite averages 32 words versus 14 for the official definitions, and no matched-length control exists for this catalog, so the comparison evaluates the boundary-aware content package rather than length in isolation.

All 15 runs completed (Qwen3-0.6B, 3 seeds); Table 10 reports the mean results in the main text.

Relation names carry the dominant transferable signal: the `P1`→`P2` rung is $+0.435/+0.243/+0.051$ at n=5/20/100, versus $+0.013/-0.013/-0.005$ for `P2`→`P3`. At n=20, boundary-aware content beats the official catalog in $3/3$ seeds ($+0.030/+0.025/+0.009$, mean $+0.021$) and turns the description effect positive relative to `P2` ($+0.007$ vs. -0.013). At n=5, supervision is the binding resource: both description variants add $+0.013$ over names and are equivalent in paired mean.

The semantic bottleneck changes with the regime. Relation names provide most transferable meaning on Wiki80; once supervision is sufficient at n=20, explicit boundary statements outperform generic definitions. At n=5, additional text cannot substitute for missing examples. The content-and-length package moves in the same direction as the CLINC150 audit (Appendix H), where explicit distinctions mark the highest-gain classes.

Table 10: **Wiki80 catalog comparison** (test Macro-F1, mean over 3 seeds). `P1` uses opaque labels; P3-off uses official Wikidata descriptions; P3-bnd uses the frozen boundary-aware rewrite.

n	`P1`	`P2`	P3-official	P3-boundary
5	0.0430	0.4780	0.4910	0.4910
20	0.4220	0.6650	0.6510	**0.6720**
100	0.7580	0.8090	0.8050	—

J Attraction and Interference Details

Long text on resolved classes can attract cross-class mispredictions, which is one reason coverage beats concentration: removing descriptions from the 38 most-resolved classes cuts false positives into them by 10.7 per seed in all three seeds and raises their held-out F1 by $+0.0070$, while a same-size random removal costs the removed classes -0.0125 F1. The attraction effect appears at 38 removed classes and does not persist at 75. Selection remains limited by reproducibility rather than signal strength (Section 4.4).

Descriptions are a training resource: each checkpoint is evaluated with the other catalog at inference in Table 11.

Table 11: **Training–inference catalog mismatch** on CLINC150 (Macro-F1 change, points).

Mismatch	$n=5$	$n=20$	$n=100$
`P2`-trained, add descriptions	-3.6	-3.1	-1.4
`P3`-trained, remove descriptions	-13.6	-9.9	-6.7

Removing descriptions for a subset of classes and retraining isolates where the removed text was helping versus hurting (Qwen3-1.7B, n=5; Table 12). FP counts are mispredictions into the classes whose description was removed; FN counts are their own errors; $\Delta F1$ is their held-out per-class change, all versus the full catalog with the same excluded set.

K Adaptation: Design, Per-Cell Results, and Cost

Design. Every adaptation run starts from the stage-1 checkpoint that produced the corresponding number in Sections 4–5, and changes only the prompt the model is trained under. Data, model,

Table 12: **Where removed text was helping or hurting** (CLINC150, Qwen3-1.7B, n=5, 3 seeds).

removal	described	FP into removed	FN of removed	$\Delta F1$ of removed
difficulty: remove 38 most resolved	112	$-\mathbf{10.7}$ (3/3 seeds)	$+3.7$	$+\mathbf{0.0070}$
random 75% (112 described)	112	-16.3 (3/3 seeds)	$+24.7$	-0.0125
difficulty: remove 75 most resolved	75	$+5.3$ (1/3 seeds)	$+3.3$	-0.0010
random 50% (75 described)	75	-8.3 (2/3 seeds)	$+19.7$	-0.0103

optimizer, batch size, and seeds are unchanged from the source run. The standard recipe is lr 2×10^{-5} with at most ten epochs and patience three, selecting the best epoch by validation Macro-F1 *under the deployment prompt*; the gentle recipe uses lr 2×10^{-6}, at most four epochs, patience one. Cells at $n \geq 20$ use at most three epochs (patience one) to bound compute; the n=5 arm early-stopped at three to four epochs, so this reproduces the observed optimum rather than extending it. Every checkpoint is evaluated under all three prompts (names / compact gloss / full) on the shared test split. The equal-budget control continues training the same checkpoint under the unchanged rich prompt for the same recipe.

Equal-training-recipe control. The "vs scratch" column of Table 14 compares the adapted model with a model trained on names from scratch, but the adapted model runs *two* stages and the baseline runs one, so that column cannot separate "the stage-1 descriptions are in the weights" from "the model trained longer". The equal-budget control of Table 16 holds the second stage's prompt fixed (rich→rich) and answers the other question. The missing control changes one field of the standard recipe: it starts from the *names*-trained stage-1 checkpoint instead of the rich one, and continues under the names prompt with the same data, seeds, optimizer, budget and model-selection rule. The two paths have the same training recipe, though their validation-selected total epochs can differ because each stage stops independently. A second variant sets the names path's second-stage epoch budget to target the rich path's selected total epochs per seed (`patience` set to the epoch budget to keep the window open; checkpoint selection remains unchanged and can choose an earlier epoch).

Table 13: **Equal-training-recipe control** (CLINC150, Qwen3-1.7B, n=5; Macro-F1). "rich→names" is the standard recipe, "control" is the same recipe from a names-trained stage-1, and "matched" sets the control's stage-2 budget per seed to target the rich path's selected total; checkpoint selection may yield fewer epochs. "realized" lists stage-1 + stage-2 selected epochs. Seed 101 is excluded from the matched variant: the names path's stage-1 alone used more epochs than the rich path's total.

seed	rich→names	control	matched	− control	− matched	realized
13	0.9167	0.9036	0.9075	$+0.0131$	$+0.0092$	$7/7/7$
42	0.9029	0.9045	0.9006	-0.0016	$+0.0023$	$10/7/7$
87	0.9129	0.9069	0.8916	$+0.0059$	$+0.0213$	$11/18/11$
101	0.9099	0.9096	—	$+0.0003$	—	$9/11/$—
202	0.9149	0.9068	0.9009	$+0.0081$	$+0.0140$	$16/12/10$
303	0.9180	0.9066	0.9040	$+0.0114$	$+0.0140$	$15/10/15$
404	0.9102	0.9044	0.9049	$+0.0058$	$+0.0053$	$16/12/10$
mean				$+0.0061$	$+0.0110$	
95% CI				$[+0.0011, +0.0112]$	$[+0.0038, +0.0182]$	
seeds > 0				$6/7$	$6/6$	

Both contrasts exclude zero: $+0.0061$ ($[+0.0011, +0.0112]$) at equal budget and $+0.0110$ ($[+0.0038, +0.0182]$) with the targeted budget, over 6/7 and 6/6 positive seeds respectively; the per-seed example-level bootstrap agrees ($+0.0062$, $+0.0111$). The control's own second stage shows no detectable gain under its own prompt (control minus the names-from-scratch model: $+0.0008$, $[-0.0022, +0.0038]$, 4/7 seeds). Two qualifications. First, equal *budget* is not equal *realized* steps, because both stage-1 runs early-stop on their own validation curves and so does the control; the realized totals are in the table, and on the three seeds where the names path trained at least as many total epochs as the rich path (13 equal, 87 and 101 longer) the rich path still leads ($+0.0131$, $+0.0059$, $+0.0003$) — a descriptive reading of the per-seed table, not a pre-registered test. Second, on Banking77 (three seeds) the same control is underpowered (-0.0017, $[-0.0141, +0.0107]$); that cell was never claimed as a significant advantage over from-scratch training (its recorded interval already

contained zero), so no claim depends on it. At the rich prompt the rich-initialized model remains ahead by $+0.0273$ ($[+0.0201, +0.0346]$, 7/7 seeds), i.e. a names stage-2 does not erase what the rich stage-1 left behind.

Per-cell grid. Δ_{sem} is the description gain recorded for the same cell in the paper's substitution tables; "loss" is names-prompt accuracy of the unadapted model minus the adapted model; "vs scratch" compares the adapted model with training on names from scratch, measured in-batch in every cell.

Table 14: **The residual-law grid.** Eight cells; three seeds each except CLINC150 1.7B n=5 (seven). All quantities are Macro-F1 *points*.

cell	Δ_{sem}	unadapted	adapted	loss	net of control	vs scratch
CLINC150 0.6B n=5	+2.52	0.7643	0.8864	12.2	10.5	+1.10
CLINC150 0.6B n=20	+0.83	0.8487	0.9407	9.2	8.5	+0.10
CLINC150 0.6B n=100	−0.01	0.9007	0.9688	6.8	—	+0.08
CLINC150 1.7B n=5	+1.07	0.8020	0.9122	11.0	9.0	+0.70
CLINC150 1.7B n=20	+0.38	0.8746	0.9487	7.4	6.2	−0.01
CLINC150 4B n=5	+0.44	0.8967	0.9477	5.1	—	+0.46
Banking77 0.6B n=5	+2.78	0.6347	0.7680	13.3	12.5	+1.23
Banking77 0.6B n=20	+1.05	0.7309	0.8614	13.1	12.4	−0.01

The linear fit over the eight cells gives $6.98 + 2.46\,\Delta_{\text{sem}}$ points (r=0.80, R^2=0.64) for the loss and r=+0.84 for "vs scratch". In every cell the adaptation gain equals the loss numerically; all seeds are positive in every cell. Adaptation does not degrade the rich prompt: on the headline cell, adapted accuracy with the full catalog is 0.9233 versus 0.9159 before adaptation.

Table 15: **What the adapted model inherits** (per-class F1, seed-averaged, by names-only recall quartile of the unadapted model; Q1 least resolved). "vs scratch" compares the names-adapted model with the names-from-scratch model.

dataset	model	Q1	Q2	Q3	Q4	vs scratch
CLINC150	from scratch	0.8228	0.8938	0.9346	0.9712	—
	adapted	0.8404	0.9035	0.9345	0.9716	+0.0176 / +0.0097 / −0.0000 / +0.0004
Banking77	from scratch	0.6194	0.7054	0.8033	0.9020	—
	adapted	0.6519	0.7197	0.8058	0.9006	+0.0326 / +0.0143 / +0.0026 / −0.0014

Table 16: **The adaptation budget curve** (CLINC150, Qwen3-1.7B, n=5, names prompt; three seeds, seven for the outline rows). Recovery is the adaptation gain as a share of the unadapted model's gap to the full catalog; the control trains under the unchanged rich prompt with the same recipe.

recipe	adapted	gain	control	recovery
lr 2×10^{-6}, $\leq$ 4 epochs	0.8147	+0.0127	0.8033	11%
lr 5×10^{-6}, $\leq$ 10 epochs	0.8647	+0.0627	0.8098	57%
lr 2×10^{-5}, $\leq$ 10 epochs	0.9122	+0.1102	0.8221	97%

The advantage over training from scratch concentrates in Q1 and tracks the per-class description benefit (r=+0.23 on CLINC150 over 150 classes, +0.27 on Banking77 over 77). The reverse direction does not work: adapting a names-trained model to the rich prompt moves +0.0046 (3 seeds, interval containing zero).

Breakeven example. One adaptation run costs 27 minutes of training on a single 80 GB accelerator at 1.7B (measured as the run's wall-clock minus its three evaluations), about 0.45 GPU-hours, against a recurring saving of 2,629 tokens per request relative to the full catalog. In the breakeven expression of Appendix L, $C_{\text{extra}} = 0.45$ GPU-hours and the denominator is the saved tokens priced by the deployment. At an illustrative \$0.10 per million prompt tokens and \$2 per GPU-hour,

the one-time cost is repaid after roughly 3,400 requests; the compact-gloss comparison (723 tokens saved) breaks even after roughly 12,000. These prices are placeholders for the deployment's own.

L TRANSFER GRIDS AND BREAK-EVEN

Measured response requires paired `P2`/`P3` checkpoints; a pilot at a smaller scale can supply the ranking for a larger target. At n=5, response rankings retain useful agreement across nearby scales (Spearman $+0.31$ from 0.6B to 1.7B, permutation p=0.0003; $+0.13$ from 0.6B to 4B). Given a pilot ranking, the target selects the top-B classes and trains once with the mixed catalog; every selector describes 38 of 150 classes, so only the decision source differs (Table 17). On 1.7B, the pilot ranking reaches $+0.0069$ over names in 6 of 7 seeds (95% CI $[+0.0012, +0.0127]$) and preserves 97% of the target-side capture (65% vs. 67%; paired difference $+0.0002$, interval includes zero); it lands $+0.0036$ above random 25%, a gap whose interval includes zero. On 4B the pilot reaches 62% capture against random's 11%, on a full-catalog gap that is itself $+0.0044$. We position pilot transfer at 4B by what it saves (the one-time measurement cost) rather than by the accuracy contrast, whose seed-level interval over three seeds ($+0.0023$, $[-0.017, +0.022]$) spans zero. A 0.6B pilot recovers most of what target-side measurement recovers at 1.7B, and at 4B it delivers a usable ranking for a fraction of the one-time cost.

Table 17: **End-to-end fixed-budget selection across target scales** (CLINC150, n=5). Each selective condition describes 25% of classes; capture is the fraction of the full-`P3` gain over `P2` on the shared-seed baseline (7 seeds at 1.7B, 7-seed baseline and 3-seed selective runs at 4B).

Target	selector / source	Macro-F1	Δ Macro-F1	capture	tokens
1.7B	`P2` names	0.9052	—	—	533
	Full `P3`	0.9159	+0.0107	100%	3,162
	Random 25%	0.9085	+0.0033	31%	1,193
	Response, 0.6B pilot	**0.9121**	+0.0069	65%	1,256
	Response, 1.7B target	**0.9124**	+0.0072	67%	1,270
4B	`P2` names	0.9431	—	—	533
	Full `P3`	0.9475	+0.0044	100%	3,162
	Random 25%	0.9436	+0.0005	11%	1,249
	Response, 0.6B pilot	**0.9458**	+0.0027	62%	1,262
	Response, 4B target	**0.9468**	+0.0037	84%	1,209

Table 18: **Cross-scale consistency** of response rankings averaged over 3 seeds. Capture projects the target full-description gap recovered by the pilot top-25%; it is omitted when that gap is zero.

n	transfer	Spearman ρ	perm. p	capture
5	0.6B → 1.7B	0.31	0.0003	70%
5	0.6B → 4B	0.13	0.019	53%
20	0.6B → 1.7B	0.19	0.012	72%
20	0.6B → 4B	0.06	0.022	75%
100	0.6B → 1.7B	0.02	0.16	—
100	0.6B → 4B	−0.04	0.88	—

Capture degrades gracefully with scale distance and disappears exactly where the full-description gap itself vanishes (the n=100 rows): the capability neighborhood has a natural boundary, since allocation is unnecessary where there is nothing to allocate.

Table 19: **Reusing response rankings across model updates** (CLINC150, n=5, Macro-F1 mean over 3 seeds). Old-25 assigns descriptions using the previous model's measured top 25%.

Model update	`P2`	random 25%	old-25%	full `P3`
Qwen2.5-1.5B → Qwen3-1.7B	0.9034	0.9087	**0.9114**	0.9160
Qwen3-4B → Qwen3.5-4B	0.9410	**0.9444**	0.9441	0.9510

Let h be the prefix-cache hit rate, and let c_{miss} and c_{hit} denote the effective per-token prices (or latency weights) for uncached and cached prefix tokens. For the 1,906-token reduction of the compact catalog relative to the full catalog, the break-even request count is

$$N_{\text{BE}} = \frac{C_{\text{extra}}}{1{,}906\,[(1-h)c_{\text{miss}} + hc_{\text{hit}}]}, \tag{7}$$

where C_{extra} is the extra one-time training cost of measurement. The uncached expression is recovered with $h{=}0$. When cache hits are cheap, the financial break-even moves outward even though the catalog still occupies context and cache capacity; we therefore report token footprint as the portable cost axis and leave the conversion to the deployment stack.

M Estimation-Budget Controls

Section 4.4 reports that measured response is the most accurate per-class signal and the least reproducible, and that at a 25% budget no signal separates from random. Because R_c is estimated from 20 validation examples per class and scored against a target with 30 test examples per class, both the signal and the target are coarse. We therefore quantify the noise in each and ask how much of the Section 4.4 null is measurement-limited. Everything below is computed from cached per-example predictions (Qwen3-1.7B, $n{=}5$) and involves no new training.

Reliability of the target. Splitting the test set into halves of 15 examples per class and recomputing the per-class gain on each half gives a within-seed split-half Pearson correlation of $+0.44$ and sign agreement of only 47%; across seeds, the per-class gain ranking agrees at $+0.25$ (7 seeds). The target a selector is scored against is therefore itself about half noise at this data budget, which caps what any selector can learn and reuse.

Reliability of the signal. The same split of the 20-example dev set gives a split-half Pearson of $+0.36$ and sign agreement of 38% for R_c; its cross-seed ranking agreement is $+0.19$, and its cross-seed predictive power (seed A's R_c against seed B's held-out per-class gain) is $+0.16$.

Dev-size sweep. Rebuilding R_c from k dev examples per class shows monotone improvement with the estimation budget but a persistent stability gap (Table 20): predictive power rises from $+0.19$ ($k{=}5$) to $+0.38$ ($k{=}20$, the full dev split; this row reproduces Table 5 exactly at 3 seeds), while cross-seed reproducibility rises only from $+0.05$ to $+0.09$ (3 seeds; $+0.13$ on 7 seeds) and stays an order of magnitude below the boundary statistic's $+0.63$. More dev data of this kind helps, but the response ranking remains model-draw-specific at every budget we can test from cached predictions; whether a 5–10$\times$ larger estimation set would close the gap is not tested here, and the deployment recommendation of Section 7 (compress by default, so that no per-class estimate is needed) does not depend on the answer.

Table 20: **Dev-size sweep for the response signal.** R_c is rebuilt from k randomly drawn dev examples per class (5 repetitions at $k < 20$); the metric is the max-rank Spearman used in Table 5. Reproducibility is the cross-seed rank agreement of R_c; predictive power is its correlation with the same seed's held-out test gain. The 3-seed columns reproduce the main-text values.

k (dev/class)	reproducibility (3 seeds)	reproducibility (7 seeds)	predictive power (3 seeds)	predictive power (7 seeds)
5	$+0.050$	$+0.055$	$+0.193$	$+0.217$
10	$+0.069$	$+0.078$	$+0.300$	$+0.315$
20	$+0.093$	$+0.129$	$+0.375$	$+0.398$

N Equivalence Test for the Frontier

Because failure to reject a difference is not evidence of equivalence, we run two one-sided tests (TOST) on the seven paired seed-level Macro-F1 differences, using a symmetric ± 0.005 margin. The margin is the run-to-run reproducibility of a single condition: the mean seed-level SD over the fifteen conditions of Table 1, 0.0049 (range 0.0025–0.0072). A difference smaller than that cannot be reproduced by retraining, so it is the natural tolerance for asking whether two catalogs are interchangeable; we report the interval first and the test second, so a reader can apply a different

tolerance. For compact gloss minus full catalog, the mean difference is -0.0017 and the 90% t interval is $[-0.0057, +0.0023]$: the lower bound crosses the margin, so equivalence is not established at this tolerance (p=0.080). For hybrid top-38 minus full catalog, the mean is $+0.0002$ and the 90% interval is $[-0.0033, +0.0038]$ (95%: $[-0.0042, +0.0047]$), establishing equivalence at the one-retraining margin (p=0.020). Table 21 gives the sensitivity curve: the hybrid's verdict flips at a margin of 0.004 and the compact gloss's at 0.006. Accordingly, the paper treats the compact gloss as the lowest-context high-utility point (84% gain recovery) and reserves the equivalence claim for the hybrid at a tolerance of one retraining; where a deployment needs a tighter tolerance, the interval, not the verdict, is the quantity to use.

Ten-seed extension. Because the seven-seed verdict sits close to its margin, we added three seeds (505/606/707) to the two conditions that carry it, fixing the procedure before looking at their results. Over ten seeds the hybrid's mean difference from the full catalog is $+0.0014$ (90% interval $[-0.0028, +0.0057]$; 95% $[-0.0038, +0.0067]$), i.e. 113% of the full-catalog gain at 58% of the tokens, and the TOST verdict at the one-retraining margin is *no longer established* (p=0.079). The interval widens on the positive side rather than shifting against the hybrid: seed 505 drew a full-catalog run that fell below its own names-only baseline (0.9032 vs. 0.9063), the regime-dependent outcome the substitution analysis of Section 4 predicts, while the hybrid at that seed reached 0.9222. We therefore report the practical bound rather than the verdict: at 58% of the tokens, the hybrid's accuracy *loss* relative to the full catalog is bounded by 0.4 points (95% one-sided) while its point estimate is above it; the main text accordingly states equivalence as a seven-seed result, reports the ten-seed difference and interval, and presents the ten-seed test as *not* establishing equivalence rather than as a verdict. Adding seeds turned a marginal pass into an interval that can be stated without a tolerance at all.

Table 21: **TOST sensitivity to the equivalence margin** (CLINC150, Qwen3-1.7B, n=5, 7 seeds; same paired seed-level procedure as Table 1). Each operating point is compared with the full catalog; the paper's margin is 0.005 (one retraining's noise). The hybrid passes from 0.004 upward, the compact gloss only from 0.006 upward.

operating point	0.002	0.003	0.004	0.005	0.006	0.007	0.010
hybrid top-38 − full	0.185	0.089	**0.042**	**0.020**	0.009	0.005	0.001
compact gloss − full	0.449	0.278	0.155	0.080	**0.041**	0.021	0.003

Stabilization does not rescue the ranking. On Banking77 at n=20 (where the aggregate catalog still gains $+0.0105$ but response selection returns -2% capture), we tried four dev-side stabilizers of the response ranking: bootstrap aggregation over resamples (seed-pair Spearman $+0.052$), shrinkage toward the stable boundary statistic ($+0.055$), rank fusion with the boundary statistic ($+0.111$), and cross-seed majority voting (self-consistent by construction, but its top-25% overlap with the test-side oracle is only 0.42, and its realized per-class gain is $+0.0228$ against the oracle's $+0.0467$ on a $+0.0105$ all-class mean). None recovers a reusable ranking, which is the operational form of the stability gate in Section 4.4.

O Estimation Budget: Five Times More Labelled Data

The dev-size sweep of Appendix M could only vary the budget downward, because the validation split holds 20 examples per class. We therefore built a larger held-out pool per seed (100 examples per class, 15,000 examples: the train-split rows the checkpoint never saw plus the validation split) and re-ran the P2/P3 checkpoints on it, inference only. Table 22 reports the same three quantities as the sweep, now up to $5\times$ the paper's dev budget.

Two readings. First, the signal scales with the estimation budget: quintupling it takes predictive power from $+0.28$ to $+0.52$, ranking reproducibility from $+0.07$ to $+0.18$, and the oracle-normalised capture of a 25% selection from 35% to 63%. Estimation budget is therefore a first-order design variable, and a practitioner with cheap labels should re-measure as the pool grows. Second, the stability ordering is unchanged: even at 100 examples per class, cross-seed reproducibility ($+0.18$) stays about $3.5\times$ below the stable boundary statistic ($+0.63$), so the stability gate and the compress-by-default recommendation apply across the dev budgets we can test. The k=20 row here is a subsample of the mixed pool and does not coincide with the validation-only 20/class anchor of

Table 22: **Response-signal quality against the estimation budget** (Qwen3-1.7B, n=5, 7 seeds; max-rank Spearman as in Table 5). "Selected gain" is the mean held-out per-class gain of the classes the R_c ranking picks at a 25% budget; capture normalises it between the all-class mean (0%) and the test-side oracle (100%).

k (dev/class)	reproducibility	predictive power	selected gain (25%)	capture
5	$+0.016$	$+0.123$	$+0.0230$	18.1%
10	$+0.012$	$+0.252$	$+0.0319$	31.4%
20	$+0.068$	$+0.283$	$+0.0344$	35.0%
50	$+0.094$	$+0.432$	$+0.0484$	55.7%
100	$\mathbf{+0.175}$	$\mathbf{+0.524}$	$\mathbf{+0.0537}$	**63.4%**

Table 23: **Per-class gain by boundary-resolution quartile** (seed-averaged; Q1 = least-resolved quartile by `P2` test recall). Description value is concentrated in Q1 at every scale tested; the full catalog's extra length does not add Q1 value and damages resolved classes in weaker models.

Setting	policy	Q1	Q2	Q3	Q4
Qwen3-0.6B n=5 (7 seeds)	full catalog	$\mathbf{+0.0692}$	$+0.0231$	$+0.0014$	-0.0007
	compact gloss	$+0.0547$	$+0.0190$	$+0.0010$	-0.0053
Qwen3-1.7B n=5 (7 seeds)	full catalog	$\mathbf{+0.0404}$	$+0.0087$	-0.0013	-0.0044
	compact gloss	$+0.0323$	$+0.0082$	-0.0028	-0.0013
Qwen3-4B n=5 (3 seeds)	full catalog	$+0.0193$	$+0.0072$	$+0.0009$	-0.0001
	compact gloss	$+0.0093$	-0.0061	-0.0051	-0.0016
Gemma-3-1B n=5 (3 seeds)	full catalog	$\mathbf{+0.1688}$	$+0.0367$	-0.0409	-0.0295
	compact gloss	$+0.1610$	$+0.0609$	-0.0076	-0.0300
Gemma-3-4B n=5 (3 seeds)	full catalog	$+0.0507$	$+0.0089$	-0.0044	-0.0038
	compact gloss	$+0.0461$	$+0.0079$	-0.0037	-0.0039

Table 5 (reproducibility $+0.13$, power $+0.40$); that spread across pools of the same nominal size is itself an instance of the estimation noise this appendix documents.

P Per-Class Gain by Boundary-Resolution Quartile

Quartiles are formed per seed from the names-only (`P2`) test recall; Q1 is the least-resolved quartile. Across the 150 classes at n=5 on the 0.6B pilot, class difficulty $D_c = 1 - \mathrm{Recall}_c$ correlates with $\Delta F1_c$ at $+0.52$ (Spearman), and symmetric confusion $C_c = \max_{j \neq c}(E_{cj} + E_{jc})/2$ at $+0.54$ ($+0.35$ after controlling for recall).

Deployable version of the split. Table 23 forms quartiles from `P2` *test* recall, which a deployment cannot read. Table 24 repeats the split with the seed's `P2` *validation* recall on the same per-class gains. The concentration survives but is attenuated: the least-resolved quartile carries $+3.05$ of the diagnostic $+4.04$ points at 1.7B (75%; 88% at 0.6B, 73% at 4B), and the ordering among the middle quartiles becomes noisier ($Q2$ and $Q3$ interchange at 0.6B and 4B). The gradient available before deployment is therefore smaller than the diagnostic gradient, consistent with the estimation noise of Appendix M and with the selector results of Appendix D; the hybrid policy in Section 5 reads precisely this weaker but deployable statistic.

Table 24: **Diagnostic versus deployable quartile split** (CLINC150, mean per-class test ΔF1 in points, seed-averaged; quartiles formed per seed, Q1 = least-resolved). Test recall is the diagnostic used in Table 23; validation recall is the statistic a selector can read before deployment.

Setting	quartile statistic	Q1	Q2	Q3	Q4
Qwen3-0.6B n=5 (3 seeds)	`P2` test recall	$\mathbf{+6.75}$	$+2.63$	$+0.27$	$+0.54$
	`P2` validation recall	$+5.97$	$+2.91$	$+0.90$	$+0.42$
Qwen3-1.7B n=5 (7 seeds)	`P2` test recall	$\mathbf{+4.04}$	$+0.87$	-0.13	-0.44
	`P2` validation recall	$+3.05$	$+0.95$	$+0.55$	-0.22
Qwen3-4B n=5 (3 seeds)	`P2` test recall	$\mathbf{+1.93}$	$+0.72$	$+0.09$	-0.01
	`P2` validation recall	$+1.41$	$+0.40$	$+0.93$	-0.02

Table 25: **Compression across regimes** (CLINC150; 7 seeds for Qwen3, 3 for Gemma-3). The compact gloss costs ≈40% of the full catalog's tokens everywhere. It matches or exceeds the full catalog in four of the five n=5 settings, and at Gemma-3-1B n=100; the exception is the regime where the full catalog itself barely adds value, and where targeted full text is the policy of choice. Capture ratios are unstable where the full-catalog gain is near zero.

setting (seeds)	P2	compact gloss	full catalog	capture	full − compact
Gemma-3-1B (3)	0.8548	**0.8999**	0.8878	**137%**	-0.0123 $[-0.0314, +0.0068]$
Gemma-3-4B (3)	0.9362	0.9476	0.9488	90%	$+0.0013$ $[-0.0007, +0.0032]$
Qwen3-1.7B (7)	0.9052	0.9142	0.9159	84%	$+0.0017$ $[-0.0033, +0.0067]$
Qwen3-0.6B (7)	0.8751	0.8921	0.8980	74%	$+0.0059$ $[-0.0012, +0.0131]$
Qwen3-4B (7)	0.9431	0.9403	0.9475	−64%	$+0.0072$ $[+0.0021, +0.0124]$
Gemma-3-1B, n=100 (3)	0.9198	**0.9615**	0.9469	**154%**	-0.0147 $[-0.0211, -0.0084]$
Gemma-3-4B, n=20 (3)	0.9659	0.9667	0.9662	290%	-0.0005 $[-0.0049, +0.0038]$

Scope of the adaptation result. It covers continued supervised training with a deployment prompt fixed in advance.

Further frontier rows. Three additional policies from the 1.7B frontier table: compact + full text top-112 (0.9142, +0.0090 over P2, 85% capture, 2,738 tokens), random 50% selective (0.9107, +0.0055, 52%, 1,837), and difficulty-selected 25 from the 0.6B pilot (0.9136, +0.0084, 79%, 1,244). They support the same ordering as the rows kept in Table 1.

Q Compression Across Regimes

The compact catalog is evaluated in five n=5 settings spanning two model families, plus two larger-data settings (Table 25).

The 4B regime in paired terms. The Qwen3-4B row is the one setting where the compact gloss alone loses value, so it carries the paper's depth-on-demand claim and we report its paired statistics explicitly (7 seeds; intervals computed as in Section 5). The gloss is 0.0028 below names-only ($[-0.0033, +0.0088]$), the full catalog beats the gloss by $+0.0072$ ($[+0.0021, +0.0124]$, 6/7 seeds), and restoring full text for the 38 least-resolved classes recovers $+0.0052$ over the gloss ($[-0.0004, +0.0108]$, 6/7 seeds). The hybrid then sits 0.0021 below the full catalog, with interval $[-0.0061, +0.0019]$: because that interval is not contained in the ± 0.005 margin, the equivalence established at 1.7B over seven seeds (Appendix N) does *not* hold at 4B. The relevant scale is the one the substitution curve predicts—at 4B the full catalog's own gain over names is $+0.0044$ ($[-0.0024, +0.0112]$), so all four policies sit inside a band narrower than the margin the equivalence test can resolve.

R The Frontier at 0.6B

The regime table reports one interior point at the weakest scale; we complete the same frontier there, where the base model leaves the most residual value (74% capture against 84% at 1.7B). Table 26 reports the full grid at 7 seeds, same protocol and same validation-based selector as Section 5 (the K least-resolved classes by the seed's own dev P2 recall).

Table 26: **The semantic-budget frontier at 0.6B** (CLINC150, Qwen3-0.6B, n=5, 7 seeds). Same protocol as Table 1; only the catalog differs.

catalog policy	Macro-F1	vs. P2	capture	tokens	gain / 1k tok
P2 names	0.8751	—	0%	533	—
compact gloss (all 150)	0.8921	+0.0171	74%	1,256	1.36e-2
compact + full text, top-38	0.8941	+0.0190	83%	1,785	1.06e-2
compact + full text, top-75	**0.8990**	**+0.0240**	**104%**	**2,292**	1.04e-2
compact + full text, top-112	**0.8997**	**+0.0246**	**107%**	2,776	8.86e-3
full catalog	0.8980	+0.0229	100%	3,162	7.26e-3

The demand point moves outward as the model weakens. Restoring full text for the 38 least-resolved classes is not distinguishable from the compact gloss alone ($+0.0020$, $[-0.0084, +0.0044]$), and top-75 beats top-38 significantly (top-75 $-$ top-38 $= +0.0050$, $[-0.0001, +0.0099]$, 7/7 seeds), reaching 104% of the full-catalog gain at 73% of its tokens. At 1.7B the same comparison ran the other way: there top-38 already matches the full catalog and top-75 is no better. The rule is Section 4's substitution read as a design instruction: the fewer boundaries data and capability have resolved, the more depth remains worth paying for, and the policy's operating point sits on the frontier's interior, not at its endpoints. Banking77 at the same scale shows the limit case (Appendix W).

S Independent Compact Catalogs

The compact catalog of the main experiments is one AI compression pass with an AI verification pass (Appendix A). To test whether the allocation results depend on that particular wording, we generated two more catalogs with independent generators and instructions, froze both before training, and ran the same 1.7B/n=5 experiment (7 seeds each). Catalog A used a different model family with the paper's instruction; catalog B used a different model and a differently worded instruction. Both prompts, the sampling seed, and a SHA-256 per catalog are recorded with the artifacts, and every gloss is capped at four words so the budget stays comparable.

Table 27: **Independently generated compact catalogs** (CLINC150, Qwen3-1.7B, n=5, 7 seeds). The verified catalog is the paper's; A and B are single unconstrained generation passes from gemma-4-31B-it and Qwen3.5-9B; C is the auditable boundary-aware recipe of the same generator as A.

catalog	Macro-F1	vs. P2	capture	tokens	vs. verified
P2 names	0.9052	—	0%	533	—
verified compact gloss	0.9142	+0.0090	84%	1,256	—
compact A (gemma-4-31B)	0.9104	+0.0051	48%	1,129	-0.0039 $[-0.0108, +0.0030]$
compact B (Qwen3.5-9B)	0.9091	+0.0038	36%	1,175	-0.0052 $[-0.0122, +0.0018]$
compact C (boundary-aware)	0.9127	+0.0075	70%	1,143	-0.0015 $[-0.0081, +0.0051]$
full catalog	0.9159	+0.0107	100%	3,162	—

An automatic, auditable recipe (catalog C). Catalogs A and B leave the recipe unspecified, so we built a third catalog that makes the paper's own instruction mechanically checkable: every gloss must contain at least one content word that appears in the class's frozen description and does "not" appear in the description of its nearest lexical neighbour (an anti-collision cue), with one regeneration attempt and an extractive fallback, all recorded per class. Generation used the same model as catalog A, so the contrast against A isolates the constraint rather than the generator. All 150 entries passed the audit (17 have no cue and are exempt; 19 needed the second attempt; no fallbacks), at 2.87 mean words and 1,143 prompt tokens. Catalog C recovers 70% of the full-catalog gain (0.9127, $+0.0075$ over names, 95% CI $[+0.0007, +0.0147]$), 0.24 points above catalog A from the same generator ($+0.0024$, $[-0.0040, +0.0090]$), and it is not distinguishable from the verified catalog (-0.0015, $[-0.0081, +0.0051]$) or from the full catalog ($+0.0032$ for full minus C, $[-0.0036, +0.0102]$) despite costing 36% of its tokens. **Reading.** All four catalogs point the same way: each lands above names, at a third of the full catalog's tokens, and no higher than the full catalog's point estimate; the two unconstrained catalogs are the weakest. The honest reading is therefore directional robustness with a recipe-dependent magnitude, and the recipe is now stated: a 2–4 word gloss that keeps a cue separating the class from its nearest neighbour recovers 70% of the full-catalog gain automatically, and the two-pass verified catalog sits at the top of the observed range (84%). The capture percentages of Section 5 should be read as properties of that recipe, with about ± 0.1 of generator-to-generator spread, rather than of compression to a given length.

Independent generation inside the full policy. The hybrid of Section 5 gives every class a gloss and restores full text at the 38 least-resolved classes. Rebuilding it with catalog C in place of the two-pass AI-verified gloss — the same 38 classes, read back from the recorded hybrids so the selection is bit-identical — tests the allocation rule under an independent frozen generation recipe. It reproduces the hybrid's accuracy and cost position, while evidence for the targeting step remains specific to the two-pass catalog. The catalog-C hybrid reaches 0.9166 at 1,737 mean prompt tokens against 0.9161 at 1,826 for the two-pass one ($+0.0005$, $[-0.0039, +0.0048]$, 7 seeds), and catalog C

alone is indistinguishable from the two-pass compact catalog (-0.0015, $[-0.0096, +0.0066]$). The two contrasts that identify *depth* on catalog C are both unresolved: adding the 38 full descriptions moves $+0.0039$ ($[-0.0034, +0.0111]$), and when the same number of classes is drawn at random the targeted hybrid leads by $+0.0019$ ($[-0.0045, +0.0083]$), against $+0.0055$ ($[+0.0010, +0.0100]$) for the two-pass catalog. We pre-registered the first of these as this batch's primary criterion and it did not pass; the targeted-versus-random control was added after seeing that result, with its own criterion registered before its runs. Independent generation thus preserves the frontier position; the targeting contrast still needs replication across recipes.

The same recipe transplanted to Banking77 unchanged (same generator, temperature, seed, anti-collision audit; 77 entries, 66 first-pass, 11 second attempt, no fallbacks, 3.16 mean words) does not reproduce the existing gloss's standalone gain: 0.7618 at 785 tokens against 0.7702 at 789 (-0.0084, $[-0.0183, +0.0015]$) and $+0.0041$ ($[-0.0105, +0.0186]$) over names, with all seven extension seeds trained here so the endpoints pair. Adding the 19 least-resolved classes in full text recovers 0.7695 at 1,061 tokens ($+0.0118$, $[+0.0012, +0.0224]$ over names, 5/7 seeds) with the depth step itself at $+0.0077$ ($[-0.0011, +0.0166]$). The recipe's standalone value is therefore dataset-sensitive, and we do not read the transfer as a cross-dataset result.

Table 28: **Catalogs built by the frozen automatic recipe, in the full policy.** CLINC150 rows are 7 seeds; Banking77 rows are 7 seeds for the four new arms and 3 for the verified hybrid (its extension seeds were not trained). Deltas are paired over seeds, same convention as Table 1.

dataset	catalog	Macro-F1	tokens	vs. names	vs. gloss
CLINC150	verified compact	0.9142	1,256	+0.0090	—
	automated C	0.9127	1,143	+0.0075	−0.0015
	verified hybrid (top-38)	0.9161	1,826	+0.0109	+0.0019
	automated hybrid (top-38)	**0.9166**	**1,737**	+0.0114	+0.0039
	automated hybrid (random 38)	0.9147	1,649	+0.0095	—
Banking77	verified compact	0.7702	789	+0.0124	—
	automated C	0.7618	785	+0.0041	−0.0084
	automated hybrid (top-19)	**0.7695**	**1,061**	+0.0118	+0.0077
	verified hybrid (top-19, 3 seeds)	0.7758	1,062	—	—

T Matched-Cost Controls: Truncation and Retrieval

T.1 Distillation versus length

The compact gloss both shortens and rewrites each description. To separate the two, we built a truncation control that keeps the first four words of each description verbatim, the point on the length axis with the same recurring cost as the gloss (1,284 versus 1,256 mean prompt tokens, a 2.2% difference), and trained the same 1.7B/n=5 models on it (7 seeds).

Table 29: **Truncation versus distillation at a matched token budget** (CLINC150, Qwen3-1.7B, n=5, 7 seeds). Both compressed catalogs cost ≈1.25k tokens, 40% of the full catalog.

catalog	Macro-F1	vs. P2	tokens
P2 names	0.9052	—	533
truncated (first 4 words)	0.9038	−0.0014	1,284
compact gloss (distilled)	**0.9142**	**+0.0090**	**1,256**
full catalog	0.9159	+0.0107	3,162

The answer is unambiguous: truncation gains nothing. The truncated catalog is statistically indistinguishable from names-only (-0.0014, $[-0.0064, +0.0036]$) and sits 1.04 points below the gloss, which wins in all seven seeds (-0.0104, $[-0.0162, -0.0046]$). The value of a compact catalog therefore comes from *which* short text each class carries, not from the reduction in length; the same conclusion the Wiki80 catalog swap reaches from the other direction (Appendix I), and a direct control for the concern that the compact catalog is merely a shorter catalog.

T.2 PER-INPUT RETRIEVAL OF DESCRIPTIONS

The static policies above give every input the same catalog. The standard alternative is per-input retrieval: select a small set of classes for each input and give only those a full description. We build that baseline with the names-only checkpoint: every class candidate is scored for every example and the top-38 are kept, with no access to test labels. We use the same mask at training and evaluation time, since descriptions are a training resource (Appendix J). The catalog costs 1,219 mean prompt tokens, between names (533) and the compact gloss (1,256), and describes the same number of classes as the fixed hybrid top-38.

Table 30: **Per-input retrieval versus static catalogs** (CLINC150, Qwen3-1.7B, n=5, 7 seeds). The retrieval mask is the top-38 classes per input under the names-only checkpoint.

catalog	Macro-F1	tokens	vs. P2
P2 names	0.9052	533	—
compact gloss (all 150)	**0.9142**	1,256	**+0.0090**
hybrid top-38 (fixed)	**0.9161**	1,826	**+0.0109**
retrieval top-38 (per input)	0.9050	1,219	−0.0002
full catalog	0.9159	3,162	+0.0107

Retrieval at this budget does not work. It lands at 0.9050, statistically indistinguishable from names-only (-0.0002, $[-0.0047, +0.0043]$, 3/7 seeds positive) and significantly below the compact gloss (-0.0092, $[-0.0170, -0.0013]$), the fixed hybrid (-0.0111, $[-0.0174, -0.0049]$, 0/7 seeds) and the full catalog (-0.0109, $[-0.0197, -0.0020]$). The losing policies share a design feature: the budget is concentrated on a minority of classes and the rest carry no text at all. Per-input selection is the most favourable setting for that concentration (the allocation adapts to each input instead of being fixed globally), and breadth still wins by a significant margin, the same direction as the truncation control (Appendix T.1) and the Wiki80 catalog swap (Appendix I).

U WRONG-PRIOR INTERVENTION

We perturb descriptions for 30 representative classes (top-10, middle-10, and bottom-10 benefit classes at n=5) using SHORT, AMBIGUOUS, and CONFLICTING variants, then retrain. The class-level response isolates boundary-sensitive effects that aggregate Macro-F1 ($\leq \pm 0.004$) averages away: ambiguity changes `change_ai_name` by -0.123; conflicting boundaries change `jump_start` by -0.137 and `how_busy` by -0.100; shortening changes `directions` by -0.067. The concentration on local boundaries reinforces class-level allocation as the relevant analysis scale.

V SELECTIVE ALLOCATION ON BANKING77

Full selective-allocation results on Banking77 (Qwen3-0.6B, test Macro-F1, mean over 3 seeds; capture: share of the full-description gap). The budget differs by row and is stated explicitly: Random, Recall and Confusion describe a fixed 25% of classes (19 of 77), whereas Response is the adaptive dev-side rule $R_c^{\text{dev}}>0$ and Oracle is its test-side counterpart $R_c^{\text{test}}>0$, each describing whatever the sign test admits (38–55 classes; the oracle is an upper bound, never a method claim):

Table 31: **Selective prior allocation on Banking77**.

n	P2	Full P3	Random	Best heuristic	Response	Oracle
5	0.7557	0.7835	0.7635 (28%)	0.7686 (46%, conf.)	**0.7702 (52%)**	0.7788 (83%)
20	0.8614	0.8720	0.8676 (58%)	0.8647 (31%, conf.)	0.8612 (−2%)	0.8656 (40%)

At n=5 the adaptive response rule recovers 52% of the full gain, above the 25%-budget heuristics (28–46%). At n=20 the same rule returns -2%, while at the matched 25% budget every heuristic

is positive (recall $+14\%$, confusion $+31\%$, random $+58\%$), an ordering that gives no evidence for any signal over random, consistent with Table 32's collapse of ranking stability.[1]

Table 32: **Banking77 response stability.** Rank ρ is mean pairwise Spearman across seed-specific dev responses; common top counts classes selected by all three seeds under the 25% budget.

n	seed rank ρ	response–gain ρ	common top	Full-P3 gain	response capture
5	0.205	0.556	4/19	+0.0278	52%
20	0.054	0.273	1/19	+0.0105	−2%

The aggregate catalog remains useful at n=20, but its class-level response is too unstable to allocate: the seed-level ranking approaches noise and only one selected class is shared across all seeds. Aggregate description value and ranking stability form separate deployment gates, explaining the different selection outcomes at n=5 and n=20.

W The Frontier on a Second Dataset

Banking77 provides a second dataset at the same scale (Qwen3-0.6B, n=5, 3 seeds) with 77 classes and longer descriptions; its endpoints and selective baselines already exist (Table 31). We add the compact gloss and the same breadth-first policy: a gloss for every class plus the full description for the K least-resolved classes by the seed's dev `P2` recall, with K=19/38/58 the 25/50/75% budget.

Table 33: **The semantic-budget frontier on Banking77** (Qwen3-0.6B, n=5, 3 seeds). Same protocol as Table 1 with the Banking77 catalog; capture is relative to the full-catalog gain (+0.0278).

catalog policy	Macro-F1	vs. `P2`	capture	tokens	gain / 1k tok
`P2` names	0.7557	—	0%	433	—
compact gloss (all 77)	0.7653	+0.0096	35%	789	1.22e-2
compact + full text, top-19	0.7758	+0.0201	72%	1,062	1.89e-2
compact + full text, top-38	0.7741	+0.0184	66%	1,347	1.37e-2
compact + full text, top-58	**0.7841**	**+0.0284**	**102%**	1,628	1.74e-2
full catalog	0.7835	+0.0278	100%	1,866	1.49e-2

Banking77 inverts the cost profile of the CLINC150 frontier. The compact gloss alone recovers only 35% of the full-catalog gain (it is significantly below the full catalog, -0.0181, $[-0.0310, -0.0056]$), against 74% at the same scale on CLINC150, so here the boundary-resolved depth is genuinely worth paying for: top-58 reaches 102% of the full-catalog gain at 87% of its tokens. The ordering, however, is the same breadth-first one, and the breadth-preserving mixture beats the same-depth selective baseline: at a 25% depth budget the hybrid reaches 0.7758 against 0.7686 for the best fixed selective heuristic (recall 0.7650, random 0.7635). With three seeds the interior contrasts are individually underpowered (top-19 − compact $+0.0105$, $[-0.0013, +0.0225]$; top-58 − top-19 $+0.0081$, $[-0.0038, +0.0199]$), so we read the table as evidence that the demand for depth grows with the residual the base model leaves, not as a second measurement of its size.

X Full Per-Seed Results

Per-seed Δ at n=5 on CLINC150: $+0.0247/+0.0244/+0.0265$ (0.6B), all paired bootstrap CIs excluding zero. At n=100: $+0.0003/-0.0003/-0.0002$, all CIs including zero. Banking77 n=5 per-seed: $+0.0364/+0.0273/+0.0197$.

Invalid-output rates: `P0` 15.9% at n=5 (0.8% at n=100); `P1`/`P2`/`P3` $\leq 0.5\%$ throughout; `P3` slightly lower than `P2` on both datasets.

[1] A per-class decomposition explains both signs. In the mixed-catalog model, the classes the adaptive rule describes gain on average ($+0.0044$ at n=20) while the undescribed classes lose (-0.0054), netting -0.0002; the test-side oracle shows the same structure but a much larger margin on its selected classes ($+0.0151$ vs. -0.0117, net $+0.0042$). Aggregate capture is therefore not a statement about whether the selected classes benefit (they do) but about whether that benefit outweighs the text removed from the rest.

An adaptive $R_c > 0$ analysis independently traces the trade-off: the selected share contracts from 81–88 classes ($\sim$55%) at n=5 to 47–58 ($\sim$35%) at n=20 and 31–35 ($\sim$22%) at n=100. The main text uses fixed budgets for controlled selector comparisons.

Y PROMPT-ONLY REGIMES

The main experiments fine-tune on the catalog. This appendix asks whether the allocation ordering survives when nothing is trained: the same prompts are given to the base Qwen3-1.7B, with the catalog either alone (zero-shot) or followed by the full n=5 demonstration set of 750 user/assistant turns in context (ICL), which matches the SFT data budget while moving it from weights to context. Prompt construction is byte-identical to the SFT conditions; each condition scores the shared 4,500-example test set with greedy decoding. Twelve runs cover {names, compact gloss, full} $\times$ {zero-shot, 3 ICL seeds}.

Table 34: **Prompt-only catalog evaluation** (base Qwen3-1.7B, CLINC150 test, no fine-tuning; ICL rows are means over seeds 13/42/87 with the full n=5 demonstration set in context). The compact gloss captures 99% of the full-catalog gain zero-shot at 40% of the tokens, and exceeds the full catalog in context while using 1.9–2.1k fewer tokens per request.

regime	catalog	Macro-F1	vs. names	capture	mean prompt tokens
zero-shot	names	0.4981	—	0%	533
	compact gloss	0.5853	+0.0872	**99%**	1,256
	full catalog	0.5865	+0.0884	100%	3,162
ICL, n=5 in context	names	0.8936	—	0%	16,562
	compact gloss	0.9001	+0.0065 (3/3 seeds)	236%	17,286
	full catalog	0.8963	+0.0027	100%	19,191

Two readings follow. First, the substitution pattern and the breadth-first ordering do not depend on fine-tuning: the catalog is worth $+8.8$ points zero-shot and $+0.3$ to $+0.7$ points in context, and the compact gloss recovers 99% of the zero-shot gain and more than all of the in-context gain. In context, the full catalog's extra length on top of 750 examples is neutral to harmful (full minus names is positive in one of three seeds, $+0.0011/-0.0018$), the same signature as Gemma-3-1B at n=100, where the demonstrations already carry the load. Second, the token saving compounds at serving: at ICL scale the compact catalog saves 1.9–2.1k prompt tokens per request against the full catalog. Invalid-output rates stay between 1.1% and 4.1% throughout.

Z LEARNED COMPRESSION: AN LLMLINGUA-STYLE BASELINE

The truncation and retrieval controls of Appendices T.1 and T.2 hold length fixed and strip content; a reviewer can still ask whether a *learned* compressor would pick better content. We implement the token-level conditional-logprob selection of LLMLingua (Jiang et al., 2023) with Qwen3-0.6B as the scorer: each frozen description is compressed by keeping the tokens the scorer finds least surprising, in order, until the class's share of the compact gloss's 585 description-token budget is reached (realized 545; 1,220 mean prompt tokens, within 3% of the gloss's 1,256). The compressed catalog is frozen (SHA-256 recorded) and trained under the identical 7-seed protocol at 1.7B/n=5.

Table 35: **Learned compression at a matched token budget** (CLINC150, Qwen3-1.7B, n=5, 7 seeds). Perplexity-selected tokens recover 6% of the full-catalog gain and lose to the compact gloss in all seven seeds.

catalog policy	Macro-F1	vs. P2	capture	tokens
P2 names	0.9052	—	0%	533
LLMLingua-style compressed	0.9059	+0.0007	6%	1,220
compact gloss (distilled)	0.9142	+0.0090	84%	1,256
full catalog	0.9159	+0.0107	100%	3,162

Learned compression does not transfer to training-time catalog allocation. The paired difference against the compact gloss is -0.0083 in all seven seeds (95% CI $[-0.0144, -0.0022]$), and its seed spread is wide (0.8856–0.9181). The mechanism mirrors the truncation control: perplexity keeps the tokens a small model predicts easily, which are the least informative for unresolved boundaries. At a matched budget, which short text each class carries matters more than how cleverly the text was shortened.