Beyond Frequency: Output-Vocabulary Pruning by Log-Mass Maximization

Egor Krasheninnikov
Amazon Web Services
egorkr@amazon.com

Sergey Ivanov
Amazon Web Services
ssivanov@amazon.com

Nikita Kozodoi
Amazon Web Services
kozodoi@amazon.com

Natalie Tarasova
Amazon Web Services
natltar@amazon.com

Aoyu Zhang
Amazon Web Services
aoyuzhan@amazon.com

Abstract

A tokenizer defines the vocabulary a language model scores at every decoding step. Output-vocabulary pruning modifies this vocabulary after training: it keeps the model and tokenizer fixed and restricts the set of output tokens scored at inference, compressing the vocabulary-sized output layer. The dominant approach, frequency-based pruning, optimizes average token coverage, which may lead to stripping the tokens that minority languages and specialized domains depend on. We address this by giving the selection problem an explicit objective: the expected log retained probability mass under the deployment distribution, which we show is equivalent to minimizing reverse KL to the teacher and is monotone submodular. We therefore select the retained tokens by greedy maximization of this objective, which inherits a $(1 - 1/e)$-approximation guarantee. Frequency pruning is its linear surrogate, optimal only when contexts agree on a common token ranking. Across synthetic and multilingual deployments, greedy outperforms both frequency- and magnitude-based pruning, with the largest gains where that agreement breaks down. Under English-dominant calibration, greedy cuts Arabic reverse KL by 0.53 nats and improves Russian generation by $+1.4$ chrF, at negligible cost to English; the same support raises speculative-decoding draft acceptance by $+5.4$pp on Arabic. Our main experiments use Qwen3.5-0.8B, and a cross-family check on Gemma-3-1B confirms the qualitative pattern.

1 Introduction

The tokenizer of a large language model (LLM) defines two vocabularies at once: an *input* vocabulary that maps raw text to the tokens the model reads, and an *output* vocabulary over which the model scores a distribution at every decoding step. A growing line of work asks how this vocabulary can be modified *after* training, without retraining from scratch, to make the model more efficient or better adapted to a deployment (Ushio et al., 2023; Bogoychev et al., 2024; Vennam et al., 2024). *Output-vocabulary pruning* is one such post-hoc modification: it keeps the model and tokenizer fixed and restricts only the set of output tokens scored at inference time.

This restriction matters because the output vocabulary is also a systems bottleneck. As models scale, the *vocabulary-sized* part of the computation, mapping hidden states to a distribution over tens or hundreds of thousands of output tokens, can dominate cost even though in many contexts only a small fraction of the vocabulary carries meaningful predictive mass. Recent work has made this point explicit: the output embedding matrix can dominate drafting time in speculative decoding (Zhao et al., 2025; Williams et al., 2026), and confidence estimation over a large vocabulary can erode the savings from early exiting (Vincenti et al., 2024). Restricting the output vocabulary is therefore a lever for

accelerating inference (Chen et al., 2023; Leviathan et al., 2023; Li et al., 2024b; Schuster et al., 2022; Elhoushi et al., 2024), but the choice of which tokens to keep is itself a tokenization decision that determines what behavior the modified model can still express.

Prior work has shown that restricting the effective vocabulary can improve memory footprint and generation speed, especially when one can exploit language-specific or task-specific structure (Bogoychev et al., 2024). At a high level, one can imagine two strategies for such a post-hoc modification. *Input-side* pruning removes tokens from the tokenizer, retokenizes the input, and hopes the approximation remains accurate, but changing the tokenizer changes the hidden states and the entire predictive process, making it difficult to reason formally about what is preserved. *Output-side* pruning keeps the model and tokenizer fixed, restricting only the set of output tokens scored at inference time. This isolates a cleaner question:

> *If one can afford to score only K output tokens at each decoding step, which K best preserve the original model's behavior?*

This question is operationally meaningful in speculative decoding, where a reduced LM head accelerates drafting (Zhao et al., 2025; Williams et al., 2026), and in early-exit systems, where pruning the output space accelerates confidence estimation (Vincenti et al., 2024).

The difficulty is that existing pruning rules are largely heuristic. Frequency-based rules, which keep the tokens that appear most often in a calibration corpus, are simple, interpretable, and often effective. Yet they leave a fundamental question unanswered: *what is the actual optimization problem being solved?* Frequency is clearly a useful signal, but it cannot capture the fact that some tokens are globally rare yet locally indispensable. A code token such as `def`, a Cyrillic subword, or a domain-specific biomedical term may carry modest marginal frequency while concentrating a large fraction of the predictive mass in the contexts where it matters. A vocabulary policy optimized only for average popularity can therefore perform poorly on the long tail of deployment contexts.

This paper develops a theory for exactly this output-side problem. We study a fixed pretrained teacher model and a deployment distribution over contexts; the model and tokenizer remain unchanged, and only the output support is compressed. Our starting point is that restricting the output space amounts to projecting the teacher's next-token distribution onto a smaller support, turning vocabulary pruning into a *support-selection problem over predictive mass*. We propose to select the retained support by maximizing the expected log retained probability mass. This choice is not ad hoc: we show it is exactly equivalent to minimizing the expected reverse KL divergence between the teacher distribution and its support-restricted renormalization. The resulting objective is sensitive to *context coverage*: the logarithm penalizes supports that leave some contexts badly undercovered, and it induces a monotone submodular set function, yielding a near-optimal greedy algorithm with a formal $(1 - 1/e)$-approximation guarantee.

This perspective also clarifies the role of frequency, which we show is the exact optimizer of a linearized surrogate: it explains a pattern that has repeatedly appeared in practice, where simple static pruning rules offer substantial speedups but are brittle when deployment shifts across domains, languages, or generation modes (Bogoychev et al., 2024; Zhao et al., 2025; Vincenti et al., 2024).

Our main contributions are as follows. We formulate output-vocabulary pruning as a support-selection problem over the teacher's next-token distributions under a deployment distribution of contexts, and identify expected log-retained-mass as the right objective: it is equivalent to minimizing the expected reverse KL divergence between the teacher and its support-restricted renormalization. The resulting objective inherits the monotone submodular structure of `log(linear)` functions familiar from sensor placement and information-theoretic subset selection (Krause et al., 2008), yielding a greedy $(1-1/e)$-approximation guarantee and an exact lazy-greedy acceleration. We show that frequency pruning is the linear surrogate of this objective, prove a *common-ranking* condition under which it is exactly optimal, and construct an example where it is strictly suboptimal. Finally, we show empirically that this gap is an equity gap: on synthetic and real multilingual distributions, greedy redistributes budget to the underserved languages that frequency starves, improving

distributional fidelity and generation quality, with the advantage confirmed downstream on speculative-decoding draft acceptance.

2 Related Work

Speculative decoding and inference acceleration. Autoregressive decoding is the primary latency bottleneck in LLM inference. Speculative decoding addresses this by drafting multiple candidate tokens in parallel and verifying them against the target model, achieving 2–4× speedups without changing the output distribution (Leviathan et al., 2023; Cai et al., 2024; Li et al., 2024a). As vocabularies grow (e.g., 128K tokens in Llama-3), the LM head that scores the full vocabulary becomes a dominant cost within the draft step itself. Recent work has begun to address this directly: FR-Spec (Zhao et al., 2025) restricts draft candidate selection to a frequency-ranked token subset, reporting that the LM head and softmax account for 62% of drafting time for large-vocabulary targets (e.g., Llama-3-8B) and delivering up to 1.24× additional speedup over EAGLE-2; VocabTrim (Goel et al., 2025) pursues a similar drafter-vocabulary reduction for memory-bound deployments; and SpecVocab (Williams et al., 2026) proposes a speculative vocabulary that decouples the draft vocabulary from the target. Our work provides the theoretical foundation for such vocabulary restrictions: the frequency ranking used by FR-Spec is the linear surrogate of a principled submodular objective, and we identify when and why it is suboptimal. The *frequency* baseline in our experiments implements FR-Spec's drafter-vocabulary policy with calibration drawn from the deployment distribution; our greedy method improves on it across heterogeneous calibration regimes.

Vocabulary pruning and compression. Static vocabulary reduction has been explored primarily in multilingual settings, where models can be compressed by removing tokens unused in the target language (Ushio et al., 2023; Bogoychev et al., 2024; Dorkin et al., 2025), and more generally for low-compute environments (Vennam et al., 2024). On the dynamic side, Vincenti et al. (2024) prune the vocabulary at inference time to accelerate confidence estimation in early-exit architectures, and recent work applies dynamic vocabulary pruning to stabilize reinforcement learning from human feedback (Li et al., 2025) and to optimize prefill-decode disaggregation (Zhang et al., 2025). These methods are effective engineering solutions but lack a unifying objective: they do not formalize what a good pruned vocabulary should optimize, nor do they provide approximation guarantees. To our knowledge, this is the first work to formulate output-vocabulary pruning as a monotone submodular maximization problem with an exact reverse-KL interpretation.

3 Submodular Optimization Framework

We formalize output-vocabulary pruning as a *support-selection* problem for a fixed pretrained language model with frozen tokenizer and hidden states: if the model may score only a budgeted subset of output tokens at inference time, which subset preserves the teacher distribution most faithfully under a deployment distribution of contexts? We introduce a log-retained-mass objective and develop its reverse-KL interpretation, submodular structure, greedy guarantee, and connection to frequency pruning over the following subsections. All proofs are provided in Appendix A.

3.1 Problem setting and the formal optimization problem

Let V denote the finite vocabulary of a pretrained autoregressive language model and let $c \in \mathcal{C}$ denote a *context* (token prefix $x_{<t}$, or equivalently its final hidden state). Write $p_c(v) := \mathbb{P}_\theta(X_t = v \mid X_{<t} = c)$ for the teacher's next-token distribution at c. The deployment environment induces a distribution μ over contexts; the optimal reduced support should reflect that mixture rather than any corpus-independent notion of importance.

To keep the objective finite we assume a fixed *fallback support* $B \subseteq V$ always retained (e.g., a byte-level fallback or small mandatory shortlist), with $M_c(\varnothing) := \sum_{v \in B} p_c(v) > 0$ for every c.

The remaining optional tokens are $U := V \setminus B$; a feasible support has the form $S_A := B \cup A$ with $A \subseteq U$, $|A| = r$.

For a selected set $A \subseteq U$, define the *retained mass*

$$M_c(A) := \sum_{v \in B} p_c(v) + \sum_{v \in A} p_c(v). \tag{1}$$

Because B is always present, $M_c(A) > 0$ for every feasible A. We can now state the central optimization problem.

Definition 3.1 (Output-support selection problem)**.** Given a deployment distribution μ over contexts, a fallback support B, and a budget r, find

$$A^\star \in \operatorname*{argmax}_{\substack{A \subseteq U \\ |A| = r}} F(A), \tag{2}$$

where $F(A) := \mathbb{E}_{c \sim \mu}\big[\log M_c(A)\big]$.

The choice of the log is not arbitrary; an exact reverse-KL interpretation justifies it in section 3.3. First, consider what the simpler linear alternative gives.

3.2 A linear warm-up: what frequency pruning actually optimizes

Define the *marginal predictive frequency* $m(u) := \mathbb{E}_{c \sim \mu}[p_c(u)]$ for each optional token $u \in U$; this is the output-side analogue of token frequency (empirical next-token counts from a calibration corpus estimate $m(u)$). Let $L(A) := \mathbb{E}_{c \sim \mu}[M_c(A)]$ be the linear retained-mass objective.

Proposition 3.2 (Linear retained mass yields frequency pruning)**.** *For any feasible $A \subseteq U$, $L(A) = \mathbb{E}[M_c(\varnothing)] + \sum_{u \in A} m(u)$. Consequently, any maximizer of $L(A)$ under $|A| = r$ is obtained by taking the r tokens with largest marginal predictive frequency $m(u)$.*

Frequency pruning maximizes *average* retained mass, offering no protection against a few contexts being covered very poorly. Our objective replaces this linear criterion with a concave one.

3.3 Reverse-KL objective and submodular structure

Reverse KL is the natural distortion. For a fixed context c and support A, the natural reduced distribution restricts the teacher to S_A and renormalizes:

$$q^*_{c,A}(v) := p_c(v)\, \mathbf{1}\{v \in S_A\} / M_c(A). \tag{3}$$

This is exactly the reverse-KL projection onto the restricted support; forward KL $D_{\mathrm{KL}}(p_c \| q)$ is infinite whenever $\mathrm{supp}(q) \subset \mathrm{supp}(p_c)$, which is the entire purpose of pruning.

Theorem 3.3 (Exact reverse-KL characterization)**.** *For each context c and feasible support A, $q^*_{c,A}$ minimizes $D_{\mathrm{KL}}(q \| p_c)$ over q supported on S_A, and the minimum value is $-\log M_c(A)$.*

This identifies the population objective

$$F(A) := \mathbb{E}_{c \sim \mu}[\log M_c(A)], \tag{4}$$

whose maximization is exactly minimization of the expected reverse KL between the teacher and its best support-restricted renormalization.

Submodularity and greedy guarantee. The marginal gain of adding a token u in context c is $\log(1 + p_c(u)/M_c(A))$; it is nonnegative (monotonicity) and decreasing in $M_c(A)$ (submodularity). A token is valuable when it has large mass *relative to how poorly the context is already covered*, the property that protects minority contexts.

Theorem 3.4 (Monotone submodularity and greedy approximation)**.** *F is monotone submodular. Greedy maximization $A_{i+1} = A_i \cup \operatorname{argmax}_u (F(A_i \cup \{u\}) - F(A_i))$ achieves $F(A_r) \geq F(\varnothing) + (1 - 1/e)(F(A^\star) - F(\varnothing))$ for any optimal $A^\star$ with $|A^\star| = r$, and no polynomial-time algorithm can do better unless* P=NP *(Nemhauser et al., 1978; Feige, 1998).*

Theorems 3.3 and 3.4 are proved in Appendix A. The empirical objective replaces $\mathbb{E}_\mu$ with a calibration average $\widehat{F}(A) = \frac{1}{n}\sum_i \log M_i(A)$, and the diminishing-returns property enables an exact lazy-greedy acceleration with the same solution as standard greedy (algorithm 1).

3.4 When does frequency work, and when does it fail?

The log objective does not discard frequency pruning; it explains it. Define the omitted mass $R_c(A) := 1 - M_c(A)$, so that $F(A) = \mathbb{E}[\log(1 - R_c(A))]$.

Theorem 3.5 (First-order approximation). *Assume $R_c(A) \le \rho < 1$ for all feasible A and c. Then*

$$F(A) = -\mathbb{E}[R_c(A)] - \mathbb{E}[\eta(R_c(A))],$$

where $\eta(r) := \sum_{k=2}^{\infty} r^k/k$ satisfies $0 \le \eta(r) \le r^2/(2(1-\rho))$. The first-order surrogate is frequency pruning (theorem 3.2).

This explains both the success and the failure mode of frequency pruning. If omitted mass is uniformly small, the second-order remainder is negligible and frequency is a good approximation. But if some contexts concentrate the omitted mass sharply, as in heterogeneous deployments, the second-order penalty matters and the log objective can prefer a different support.

Under a strong structural condition, frequency is not merely a surrogate but the exact solution.

Theorem 3.6 (Common-ranking condition). *Assume there exists a total order $u_1 \succ \cdots \succ u_{|U|}$ on U such that $p_c(u_1) \ge \cdots \ge p_c(u_{|U|})$ for every context c. Then $A_{\text{rank}} := \{u_1, \ldots, u_r\}$ maximizes $F(A)$ over all $|A| = r$, and coincides with the top-r set by marginal frequency.*

Frequency pruning is therefore not "wrong": it is exactly optimal when all contexts agree on a token ranking. What breaks in heterogeneous deployments is the common-ranking assumption itself. Once contexts disagree on which tokens matter, as they do when a deployment mixes code, natural language, and specialized domains, frequency can fail.

That failure is not merely theoretical:

Theorem 3.7 (Frequency pruning can be strictly suboptimal). *There exist a deployment distribution over contexts and a budget r such that the support maximizing $\sum_{u\in A} m(u)$ is not a maximizer of $F(A)$.*

The construction (in Appendix A) uses two equiprobable contexts with complementary specialist tokens: a token u that is slightly more frequent globally, which the linear criterion keeps, and tokens b, c that each rescue a different context, which the log objective prefers for their coverage diversity. This is the mechanism behind section 4: in multi-domain deployments frequency spends the budget on majority-domain tokens, while greedy distributes it across all domains.

The theory concerns a fixed global support chosen offline; per-context online selection is trivial when the full teacher distribution is available. The fallback support B is itself part of the design space, any always-available support that keeps the reverse-KL objective finite.

4 Experiments

We evaluate greedy log-retained-mass selection (algorithm 1) against frequency pruning along the axis the theory predicts matters: calibration heterogeneity. Across a synthetic benchmark (section 4.1), a multilingual distributional study (section 4.2), text continuation (section 4.3), a robustness suite (section 4.4), and speculative-decoding draft acceptance (section 4.5), greedy wins where heterogeneity is high, ties under common-ranking, and concentrates gains on minority languages at negligible majority cost. We use fallback B with $\sum_{v\in B} p_c(v) = \beta = 0.01$, write K for the support size in real-data experiments (= theoretical r), and report setup details in Appendix C.

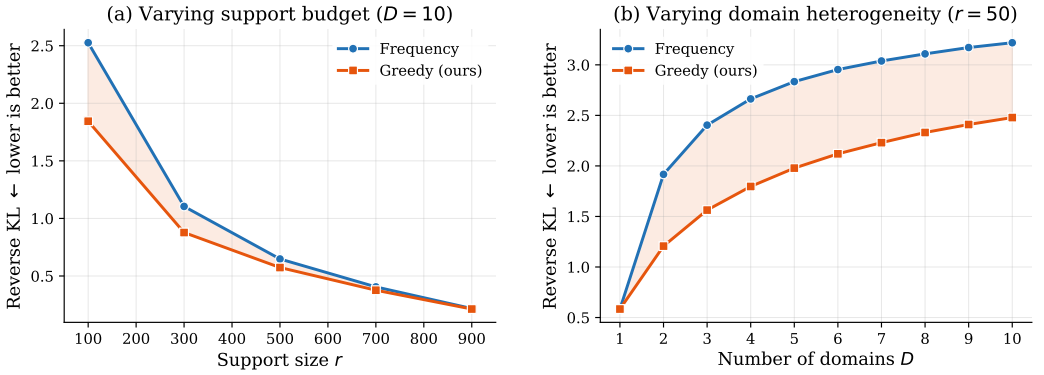


Figure 1: Synthetic multi-domain experiment. Reverse KL (lower is better) for frequency pruning versus greedy (algorithm 1). **(a)** Support budget sweep (D=10 domains): greedy consistently outperforms frequency, with the largest gap at tight budgets. **(b)** Domain heterogeneity sweep (r=50): at D=1 both methods coincide; for $D \geq 2$, greedy rescues minority domains that frequency ignores. Shaded regions indicate the improvement.

4.1 Synthetic multi-domain experiment

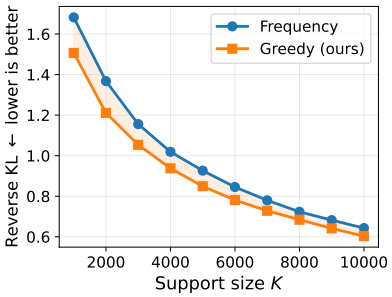
We construct a synthetic vocabulary of $|V|$=10,000 tokens with up to D=10 domains, each contributing a set of 50 domain-critical ("hot") tokens. Contexts are sampled from Dirichlet-perturbed domain prototypes, and domains are mixed with *harmonic weights* $w_d \propto 1/(d+1)$, so that the majority domain accounts for ∼34% of contexts while the smallest contributes only ∼3%. We sample 5,000 calibration and 2,500 evaluation contexts per configuration. Full details of the data generation procedure are in Appendix C.1.

In the support-size sweep (fig. 1a), we fix D=10 and vary the support size $r \in \{100, 300, 500, 700, 900\}$. Greedy consistently achieves lower reverse KL than frequency across all support sizes. The gap is largest when the budget is tight: at r=100, greedy achieves 1.84 versus 2.53 for frequency, a reduction of 0.69. At this budget, frequency leaves 49% of evaluation contexts with retained mass at the fallback floor β=0.01, meaning those contexts are effectively uncovered, while greedy reduces this fraction to 31%. As r grows and the budget suffices to cover all domain-critical tokens, the methods converge, consistent with the common-ranking condition (theorem 3.6).

In the domain-heterogeneity sweep (fig. 1b), we fix r=50 and vary $D \in \{1, \ldots, 10\}$. At D=1, both methods coincide, the single-domain setting satisfies the common-ranking condition and frequency is optimal (theorem 3.6). For $D \geq 2$, a substantial gap emerges and widens monotonically: at D=10, greedy achieves 2.48 versus 3.22 for frequency. The mechanism is precisely the one identified by theorem 3.7: with harmonic mixing weights, frequency allocates the entire budget to the one or two most prevalent domains, completely ignoring minority domains. Greedy distributes tokens roughly in proportion to domain weights, ensuring that even the smallest domain receives representation.

4.2 Multilingual distributional experiment

We validate the theory on real model distributions using Qwen3.5-0.8B-Base (Qwen Team, 2026), a language model with a vocabulary of 248,320 tokens and native support for over 200 languages. We extract next-token distributions on the FLORES-200 benchmark (NLLB Team, 2022) across 10 typologically diverse languages spanning five scripts (Latin, Cyrillic, Arabic, Devanagari, CJK). For each language, we construct ∼1,000 calibration and ∼1,000 evaluation contexts from sentence prefixes, yielding ∼10,000 contexts per split. We compare two calibration regimes: *uniform* weighting (all languages equal) and *English-dominant* weighting (60% English, 40% split among the remaining nine languages). Evaluation is always uniform across languages. Full details are in Appendix C.2.



Regime	Method	en	fr	ar	ja	ru	zh	All
Uniform	Frequency	**1.27**	1.32	1.57	1.86	1.87	2.16	1.36
	Greedy	1.28	**1.31**	**1.51**	**1.65**	**1.70**	**1.87**	**1.33**
Weighted	Frequency	**0.79**	1.35	2.39	2.34	2.42	2.73	1.68
	Greedy	0.87	**1.26**	**1.86**	**2.03**	**2.17**	**2.26**	**1.51**

Figure 2: Multilingual output-vocabulary pruning on Qwen3.5-0.8B-Base with FLORES-200 (10 languages, ∼10K contexts). *Left:* Reverse KL vs. support size K under English-dominant calibration. *Right:* Per-language reverse KL at K=1,000 (lower is better; bold indicates the better method). Under weighted calibration, frequency starves minority-script languages while greedy rescues them.

Under English-dominant calibration, greedy consistently outperforms frequency across all support sizes (fig. 2, left), with the largest gap at K=1,000 (0.17). The per-language breakdown (fig. 2, right) reveals the mechanism: frequency allocates most of the budget to English-relevant tokens, leaving non-Latin-script languages severely undercovered. Arabic reaches a reverse KL of 2.39 and Chinese 2.73 under frequency, while greedy reduces these by 0.53 and 0.47 respectively, at a cost of only 0.08 to English. Under uniform calibration, the gap is smaller (0.03 globally) but greedy still improves coverage for Chinese (-0.29), Japanese (-0.21), and Russian (-0.17). A calibration-skew sweep (Appendix D.2) confirms this monotonically, with the gap growing from $+0.13$ nats at en$=0.50$ to $+0.41$ at 0.99. Gemma-3-1B (Gemma Team, 2025) preserves the qualitative pattern at K=10,000 (Appendix D.1).

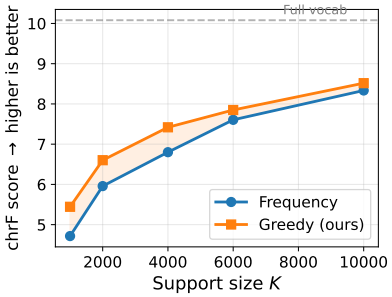
4.3 Multilingual text continuation

Distributional fidelity should translate into better generation, so we complement the previous analysis with a downstream experiment on the same model and data: we take the first 60% of each evaluation sentence as a prompt, generate up to 80 tokens via greedy decoding under the pruned vocabulary, and score with chrF (Popović, 2015), a character-level F-score that handles all scripts uniformly. We keep the English-dominant calibration (60% English) where section 4.2 showed the largest advantage and reuse the same support sets (Appendix C.3).

Greedy consistently outperforms frequency in macro-averaged chrF (fig. 3, left), with the largest gap at small K: at K=1,000, greedy achieves 5.4 versus 4.7 (+15%). The per-language breakdown (fig. 3, right) mirrors the distributional experiment. The effect is most striking on Russian: 4.4 vs. 3.0 chrF at K=1,000 (+1.4 chrF, +47% relative; full-vocab ceiling is 12.5). At K=1,000 both methods produce low-quality generation on non-Latin scripts and usable absolute quality emerges only at $K\geq$4,000; the pattern of greedy rescuing Cyrillic, Arabic, and CJK tokens that frequency discards is consistent across K.

4.4 Robustness across deployment types

The preceding experiments vary heterogeneity along a single axis (script and language mix). To test whether the theory's prediction holds more broadly, we evaluate at a fixed support size of K=1,000 on four deployments that span low to high heterogeneity: a single-script setting (5 Latin-script languages), two multi-script settings (10 languages across 5 scripts, and 5 disjoint scripts), and an extreme bilingual skew (English and Arabic at a 0.99 mixing weight). Beyond frequency, we add a data-free baseline, *magnitude* (Han et al., 2015), which keeps the K tokens with largest L_2 norm of their `lm_head` row and uses no calibration data, serving as a lower bound that isolates the value of calibrating to the deployment. Calibration follows each deployment's own label composition (Appendix C.4), and we report reverse



K	Method	en	fr	ar	ja	ru	zh	Avg
–	Full vocab	13.4	14.5	10.6	5.1	12.5	3.9	10.0
1k	Freq.	**10.1**	9.2	4.2	0.8	3.0	0.9	4.7
	Greedy	10.0	**9.7**	**6.3**	**1.0**	**4.4**	**1.1**	**5.4**
4k	Freq.	**11.3**	10.9	8.5	1.8	6.7	1.4	6.8
	Greedy	11.2	**11.3**	8.5	**2.3**	**9.1**	**1.8**	**7.4**
10k	Freq.	**12.8**	**11.8**	9.4	3.3	10.3	2.2	8.3
	Greedy	12.7	11.7	9.4	**3.6**	**10.8**	**2.5**	**8.5**

Figure 3: Multilingual text continuation on Qwen3.5-0.8B-Base with FLORES-200 under English-dominant calibration. *Left:* Macro chrF vs. support size K. *Right:* Per-language chrF at selected K values (bold indicates the better method). Greedy improves non-Latin-script languages (ru, ja, zh) at negligible cost to English.

KL and macro-averaged chrF over each deployment's languages. Full dataset definitions and baseline details are in Appendix C.4.

Table 1 shows a consistent ordering: full vocab (the ceiling) $<$ greedy $<$ frequency $\ll$ magnitude on reverse KL, and greedy $\geq$ frequency $>$ magnitude on chrF, on every deployment. The data-free magnitude baseline collapses everywhere (reverse KL above 3.4 nats, chrF at most 2.5), confirming that calibration to the deployment distribution is essential and that the contest is between the two data-driven rules. Between greedy and frequency, the gap tracks heterogeneity as the theory predicts. In the single-script Latin setting, where contexts share a common token ranking, greedy and frequency nearly coincide (1.21 vs. 1.22 nats), as theorem 3.6 requires. Where rankings diverge across scripts or under extreme skew, the gap is large: the high en+ar skew (1.55 vs. 1.95 nats, chrF 8.9 vs. 6.9), the 10-language multi-script setting (1.53 vs. 1.71), and the 5-script setting (1.25 vs. 1.32). Greedy never loses and wins when the common-ranking condition fails, the empirical signature of theorem 3.7.

4.5 Speculative-decoding draft acceptance

The previous metrics measure distributional fidelity and generation quality. We turn to the downstream metric the theory predicts: *sample-based speculative-decoding draft acceptance*. Reverse KL of the kept set under the target equals the negative log of the expected accept probability under Leviathan-Chen sampling (Leviathan et al., 2023; Chen et al., 2023), so a lower reverse KL must translate into a higher accept rate.

We reuse the four deployments of section 4.4 and measure spec-decode acceptance on the same matrix. Qwen3.5-0.8B-Base serves as both target and drafter; hard-masking the drafter's `lm_head` to the K=1,000 tokens of each method isolates the effect of support choice

	Reverse KL ↓ (nats)				chrF ↑			
Deployment	Full	Magn.	Freq.	Greedy	Full	Magn.	Freq.	Greedy
Latin (5 langs)	0	+3.50	+1.22	**+1.21**	18.6	2.5	10.8	**11.9**
Multi-script (10 langs)	0	+3.78	+1.71	**+1.53**	12.1	1.5	5.9	**6.6**
Multi-script (5 langs)	0	+3.84	+1.32	**+1.25**	14.0	0.5	7.3	**7.7**
High skew (en+ar, 0.99)	0	+3.86	+1.95	**+1.55**	13.9	1.2	6.9	**8.9**

Table 1: Robustness across deployment types on Qwen3.5-0.8B-Base at budget K=1,000, reporting reverse KL (lower is better) and chrF (higher is better). Methods: *Full*, the unpruned 248k-token ceiling; *Magn.*, a data-free `lm_head`-norm baseline; *Freq.*, frequency pruning; and *Greedy*, ours. Bold marks the best *pruned* method per metric. Greedy dominates every pruned method on every deployment, with the largest margin over frequency where heterogeneity is highest and near-parity where a common token ranking holds (theorem 3.6).

Method	en	fr	es	de	ru	ar	zh	hi	ja	sw	All
Frequency	**0.56**	0.46	0.45	0.35	0.25	0.23	0.20	**0.57**	0.22	0.39	0.37
Greedy	0.55	**0.49**	**0.46**	**0.37**	**0.28**	**0.28**	**0.27**	0.46	**0.26**	**0.40**	**0.38**
Δ pp	-0.9	$+3.1$	$+1.1$	$+2.2$	$+3.2$	$+5.4$	$+6.6$	-11.6	$+3.8$	$+1.3$	$+$**1.4**

Table 2: Per-language sample-based speculative-decoding draft acceptance at K=1,000 on the 10-language multi-script deployment of table 1 (FLORES-200, n=200, Qwen3.5-0.8B-Base). Bold marks the better method per language. Greedy improves every non-Latin-script language except Hindi ($+3.2$ to $+6.6$pp) and ties on English; the global gap is $+1.4$pp (p=0.004). Hindi (-11.6pp) is the Qwen tokenizer-specific reversal also seen in section 4.2. On the high-skew bilingual cell, greedy gains $+3.3$pp aggregate ($p<10^{-3}$) at a statistical tie on English. The single-script Latin and 5-script deployments are within bootstrap noise ($|\Delta| \leq 0.9$pp); the full aggregate matrix, draft-length sensitivity, and magnitude baseline are in Appendix D.3.

from model and tokenizer changes. At each cycle the drafter proposes γ=4 tokens that the target verifies in one forward pass, accepting with probability $\min(1, p/q)$ and resampling from $(p-q)_+$ on rejection. We report per-token acceptance on FLORES-200 prompts (n=50–200 per cell), with stratified paired-bootstrap 95% CIs on Δ = greedy $-$ frequency.

Table 2 shows per-language acceptance on the 10-language multi-script deployment. Greedy improves every non-Latin-script language except Hindi ($+3.2$ to $+6.6$pp) and ties on English, mirroring section 4.3; global gap $+1.4$pp (p=0.004). The high-skew bilingual cell shows this more sharply at the aggregate level: greedy gains $+3.3$pp ($p<10^{-3}$, paired stratified bootstrap, n=80), robust to draft length $\gamma \in \{2, 4, 8\}$ (Appendix D.3). Because this cell is 99% English, the Arabic slice is small-sample; its point estimate ($\sim +7$pp at a statistical tie on English) is consistent with the aggregate but we report only the aggregate with a confidence interval. The remaining two deployments of table 1 have reverse-KL gaps of ≤ 0.08 nats and acceptance differences within bootstrap noise (Appendix D.3). On Gemma-3-1B (Appendix D.1) greedy improves Hindi reverse KL by 0.10 nats, indicating the Hindi reversal is specific to Qwen's Devanagari subword segmentation rather than the algorithm. Our *frequency* baseline implements the same drafter-vocabulary restriction as FR-Spec (Zhao et al., 2025), with calibration drawn from the deployment distribution rather than a fixed pretraining-style corpus; greedy is a direct improvement on this axis. The acceptance gap and the underlying reverse-KL gap correspond monotonically across the four deployments: deployments where greedy ties frequency are those where theorem 3.7 predicts a tie, and the largest reverse-KL gap produces the largest acceptance gap.

5 Conclusion

This paper introduces an explicit objective for output-vocabulary pruning. We show that maximizing expected log retained mass under the deployment distribution is equivalent to minimizing reverse KL to the teacher, and that the resulting set function is monotone submodular: greedy admits a $(1-1/e)$ approximation guarantee, while frequency pruning is its linear surrogate, optimal under a common-ranking condition and suboptimal otherwise. Across four deployments on Qwen3.5-0.8B, the empirical greedy-frequency gap tracks the underlying reverse-KL gap monotonically on every metric we test, and a cross-family check on Gemma-3-1B confirms the pattern; the gains land on the minority scripts that frequency starves. Under English-dominant calibration greedy cuts Arabic reverse KL by 0.53 nats and improves Russian generation by $+1.4$ chrF, and the same support raises speculative-decoding acceptance by $+5.4$pp on Arabic ($+3.3$pp aggregate under high skew), all at negligible cost to English. A calibration-aware output-vocabulary policy is thus a drop-in replacement for the frequency rule in systems such as FR-Spec, with the largest wins where standard pruning silently underserves minority users.

Limitations

Several limitations scope our results. Our theory and experiments restrict only the *output* tokens scored at inference, holding the tokenizer and hidden states fixed; this is exactly what makes support restriction a reverse-KL projection, but it also means we do not address input-side or full tokenizer pruning, where retokenizing the prompt changes the hidden states and the entire predictive process. We likewise commit to a single global support chosen offline and apply it unchanged at every step; per-context selection is trivial when the full teacher is observable online, but dynamic adaptation of the support to the current prefix is left to future work. The efficiency gain is correspondingly confined to the output layer, reducing its cost from $O(|V|d)$ to $O((|B| + r)d)$, and we report distributional and generation-quality proxies (reverse KL, chrF, draft acceptance) rather than end-to-end wall-clock latency, which depends on KV-cache, batching, and serving-stack details outside our scope.

Our empirical validation is also deliberately controlled. Most experiments use Qwen3.5-0.8B-Base with FLORES-200, complemented by a Gemma-3-1B cross-family check, a single fallback mass $\beta = 0.01$, and a top-8,192 sparse representation of each calibration distribution that both methods share, so the truncation cannot bias the greedy-versus-frequency comparison. We compare only methods that share an output-vocabulary-only intervention with the same calibration data and budget; systems such as VocabTrim and SpecVocab combine drafter-vocabulary restriction with orthogonal architectural changes and are not numerically comparable on this axis. Generalization to other model families and vocabulary sizes, a sensitivity sweep over β, a denser calibration representation, and downstream tasks beyond continuation (code, dialogue, domain-specific corpora) all remain to be tested.

Ethics Statement

Output-vocabulary pruning changes which tokens a deployed model can still produce, so the choice of selection rule has a direct equity dimension. Frequency pruning calibrated on a majority-dominated corpus allocates its budget to the tokens of the dominant language or domain and can strip away the script-specific tokens that minority languages depend on, degrading service for exactly the users who are already underserved. Our experiments make this failure mode concrete: under English-dominant calibration, frequency pruning leaves non-Latin-script languages such as Russian, Arabic, and Chinese far less faithfully covered than English. The log-retained-mass objective is designed to counter this effect, distributing budget so that minority contexts retain usable probability mass, and we view this fairness-by-construction property as the main societal benefit of the method. We caution, however, that pruning of any kind reduces a model's expressive range, that our evaluation covers ten languages rather than the full long tail of the world's writing systems, and that practitioners should validate coverage on their own deployment distribution before pruning a production model. All data used in this work (FLORES-200) is publicly available and contains no personally identifiable information.

References

Nikolay Bogoychev, Pinzhen Chen, Barry Haddow, and Alexandra Birch. The ups and downs of large language model inference with vocabulary trimming by language heuristics. In *Proceedings of the 1st Workshop on Insights from Negative Results in NLP*, 2024.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D Lee, Deming Chen, and Tri Dao. Medusa: Simple LLM inference acceleration framework with multiple decoding heads. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 5209–5235. PMLR, 2024.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

Aleksei Dorkin, Taido Purason, and Kairit Sirts. Prune or retrain: Optimizing the vocabulary of multilingual models for Estonian. *arXiv preprint arXiv:2501.02631*, 2025.

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, Ahmed A Aly, Beidi Chen, and Carole-Jean Wu. LayerSkip: Enabling early exit inference and self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Uriel Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4): 634–652, 1998.

Gemma Team. Gemma 3 technical report, 2025.

Raghavv Goel, Sudhanshu Agrawal, Mukul Gagrani, Junyoung Park, Yifan Zao, He Zhang, Tian Liu, Yiping Yang, Xin Yuan, Jiuyan Lu, et al. Vocabtrim: Vocabulary pruning for efficient speculative decoding in llms. *arXiv preprint arXiv:2506.22694*, 2025.

Song Han, Jeff Pool, John Tran, and William J Dally. Learning both weights and connections for efficient neural networks. *Advances in Neural Information Processing Systems*, 28, 2015.

Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9, 2008.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*. PMLR, 2023.

Yingru Li, Jiawei Xu, Jiacai Liu, Yuxuan Tong, Ziniu Li, Tianle Cai, Ge Zhang, Qian Liu, and Baoxiang Wang. Dynamic vocabulary pruning: Stable LLM-RL by taming the tail. *arXiv preprint arXiv:2512.23087*, 2025.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*, 2024a.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE-2: Faster inference of language models with dynamic draft trees. *arXiv preprint arXiv:2406.16858*, 2024b.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1): 265–294, 1978.

NLLB Team. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*, 2022.

Maja Popović. chrF: Character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pp. 392–395, 2015.

Qwen Team. Qwen3.5: Towards native multimodal agents. https://qwen.ai/blog?id=qwen3.5, February 2026.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Asahi Ushio, Yi Zhou, and Jose Camacho-Collados. An efficient multilingual language model compression through vocabulary trimming. In *Findings of the Association for Computational Linguistics: EMNLP*, 2023.

Sreeram Vennam, Anish Joishy, and Ponnurangam Kumaraguru. LLM vocabulary compression for low-compute environments. *arXiv preprint arXiv:2411.06371*, 2024.

Jort Vincenti, Karim Abdel Sadek, Joan Velja, Matteo Nulli, and Metod Jazbec. Dynamic vocabulary pruning in early-exit LLMs. *arXiv preprint arXiv:2410.18952*, 2024.

Miles Williams, Young D Kwon, Rui Li, Alexandros Kouris, and Stylianos I Venieris. Speculative decoding with a speculative vocabulary. *arXiv preprint arXiv:2602.13836*, 2026.

Hao Zhang, Mengsi Lyu, Zhuo Chen, Xingrun Xing, Yulong Ao, and Yonghua Lin. PDTrim: Targeted pruning for prefill-decode disaggregation in inference. *arXiv preprint arXiv:2509.04467*, 2025.

Weilin Zhao, Tengyu Pan, Xu Han, Yudi Zhang, Ao Sun, Yuxiang Huang, Kaihuo Zhang, Weilun Zhao, Yuxuan Li, Jianyong Wang, Zhiyuan Liu, and Maosong Sun. FR-spec: Accelerating large-vocabulary language models via frequency-ranked speculative sampling. *arXiv preprint arXiv:2502.14856*, 2025.

A Proofs

This appendix collects the proofs of all formal results stated in section 3. Each proof is self-contained and preceded by a restatement of the result for the reader's convenience. We present them in the order the corresponding results appear in the main text.

A.1 Proof of theorem 3.2 (Linear retained mass yields frequency pruning)

Proposition (Restatement of theorem 3.2)**.** For any feasible $A \subseteq U$, $L(A) = \mathbb{E}[M_c(\varnothing)] + \sum_{u \in A} m(u)$. Consequently, any maximizer of $L(A)$ under $|A| = r$ is obtained by taking the r tokens with largest marginal predictive frequency $m(u)$.

Proof. From the definition of retained mass (eq. (1)),

$$M_c(A) = \sum_{v \in B} p_c(v) + \sum_{u \in A} p_c(u).$$

Taking expectation over $c \sim \mu$ and using linearity of expectation:

$$\begin{aligned} L(A) &= \mathbb{E}\Big[\sum_{v \in B} p_c(v)\Big] + \sum_{u \in A} \mathbb{E}[p_c(u)] \\ &= \mathbb{E}[M_c(\varnothing)] + \sum_{u \in A} m(u). \end{aligned}$$

The first term $\mathbb{E}[M_c(\varnothing)]$ depends only on the fallback support B and is independent of the choice of A. Therefore, maximizing $L(A)$ subject to $|A| = r$ is equivalent to maximizing $\sum_{u \in A} m(u)$, which is a modular (additive) objective. Its maximum is achieved by selecting the r tokens with the largest values of $m(u)$. □

A.2 Proof of theorem 3.3 (Exact reverse-KL characterization)

Theorem (Restatement of theorem 3.3)**.** Fix a context c and a feasible support A. Then $q^*_{c,A}$ defined in eq. (3) minimizes $D_{\mathrm{KL}}(q \| p_c)$ over all $q \in \mathcal{Q}_A$. Moreover, the minimum value is $D_{\mathrm{KL}}(q^*_{c,A} \| p_c) = -\log M_c(A)$.

The proof proceeds by a direct algebraic decomposition of the KL divergence. The key step is to introduce the restricted renormalization $q^*_{c,A}$ as an intermediate reference distribution, which separates the KL into a term we can minimize and a constant that depends only on the support.

Proof. Take any $q \in \mathcal{Q}_A$. Because q is supported on S_A, we can write

$$D_{\mathrm{KL}}(q \| p_c) = \sum_{v \in S_A} q(v) \log \frac{q(v)}{p_c(v)}.$$

For $v \in S_A$, eq. (3) gives $q^*_{c,A}(v) = p_c(v)/M_c(A)$. We can therefore rewrite the log-ratio as

$$\log \frac{q(v)}{p_c(v)} = \log \frac{q(v)}{q^*_{c,A}(v) \cdot M_c(A)} = \log \frac{q(v)}{q^*_{c,A}(v)} - \log M_c(A).$$

Substituting back into the KL expression:

$$D_{\mathrm{KL}}(q\|p_c) = \sum_{v \in S_A} q(v) \log \frac{q(v)}{q^*_{c,A}(v)} - \log M_c(A) \sum_{v \in S_A} q(v).$$

Since q is a probability distribution supported on S_A, we have $\sum_{v \in S_A} q(v) = 1$. Therefore

$$D_{\mathrm{KL}}(q\|p_c) = D_{\mathrm{KL}}(q\|q^*_{c,A}) - \log M_c(A).$$

The first term is a KL divergence, which is nonnegative and equals zero if and only if $q = q^*_{c,A}$. The second term $-\log M_c(A)$ is a constant that does not depend on q. Therefore the minimum over $q \in \mathcal{Q}_A$ is achieved at $q = q^*_{c,A}$, and the minimum value is $-\log M_c(A)$. □

A.3 Proof of theorem 3.4 (Monotonicity and submodularity)

Theorem (Restatement of theorem 3.4)**.** For every context c, the set function $f_c(A) = \log M_c(A)$ is monotone and submodular on subsets of U. Consequently, the population objective $F(A) = \mathbb{E}[f_c(A)]$ is also monotone and submodular.

The proof relies on two observations: (1) adding a token can only increase retained mass, which gives monotonicity; and (2) the map $x \mapsto \log(1 + p/x)$ is decreasing, which formalizes the diminishing-returns property that defines submodularity.

Proof. Fix a context c. Let $A \subseteq U$ and let $u \in U \setminus A$. The marginal gain of adding u is

$$\begin{aligned} &f_c(A \cup \{u\}) - f_c(A) \\ &\quad = \log\big(M_c(A) + p_c(u)\big) - \log M_c(A) \\ &\quad = \log\Big(1 + \frac{p_c(u)}{M_c(A)}\Big). \end{aligned}$$

Because $p_c(u) \geq 0$ and $M_c(A) > 0$, this quantity is nonnegative. Hence f_c is monotone: adding any token to the support can only increase (or maintain) the objective.

For submodularity, let $A \subseteq A' \subseteq U$ and let $u \in U \setminus A'$. Since $A \subseteq A'$, we have $M_c(A) \leq M_c(A')$. The function $x \mapsto \log(1 + p_c(u)/x)$ is strictly decreasing on $(0, \infty)$ (its derivative with respect to x is $-p_c(u)/(x(x + p_c(u))) \leq 0$). Therefore

$$\log\Big(1 + \frac{p_c(u)}{M_c(A)}\Big) \geq \log\Big(1 + \frac{p_c(u)}{M_c(A')}\Big),$$

which can be rewritten as

$$f_c(A \cup \{u\}) - f_c(A) \geq f_c(A' \cup \{u\}) - f_c(A').$$

This is exactly the diminishing-returns condition that defines submodularity: the marginal value of adding u is at least as large when the current set is smaller.

Finally, both monotonicity and submodularity are preserved under nonnegative linear combinations. Since $F(A) = \mathbb{E}_{c \sim \mu}[f_c(A)]$ is an expectation (i.e., a nonnegative weighted sum) of monotone submodular functions, F is itself monotone and submodular. □

A.4 Hardness and greedy approximation (part of theorem 3.4)

Theorem (Greedy guarantee, part of theorem 3.4)**.**

(i) Maximizing a monotone submodular function under a cardinality constraint is NP-hard.
(ii) No polynomial-time algorithm achieves a ratio better than $(1-1/e)$ unless P=NP.
(iii) Greedy achieves this bound:

$$F(A_r) \geq F(\varnothing) + (1 - \tfrac{1}{e})\big(F(A^\star) - F(\varnothing)\big).$$

Parts (i) and (ii) are classical results in combinatorial optimization. The NP-hardness and $(1 - 1/e)$ inapproximability follow from Feige (1998); part (iii) is the celebrated greedy guarantee of Nemhauser et al. (1978). We include the self-contained proof of part (iii) for completeness, as it is central to the paper.

Proof of part (iii). Define the normalized objective $H(A) := F(A) - F(\varnothing)$. Then $H(\varnothing) = 0$, and H remains monotone and submodular (since adding a constant preserves both properties).

Fix an iteration $i \in \{0, \ldots, r-1\}$. We will show that the gap $H(A^\star) - H(A_i)$ shrinks by a factor of at least $(1 - 1/r)$ at each step.

Since H is monotone, $H(A^\star) \leq H(A^\star \cup A_i)$. Therefore

$$H(A^\star) - H(A_i) \leq H(A^\star \cup A_i) - H(A_i).$$

By submodularity, the right-hand side can be bounded by summing individual marginal gains:

$$\begin{aligned} &H(A^\star \cup A_i) - H(A_i) \\ &\quad \leq \sum_{u \in A^\star \setminus A_i} \big(H(A_i \cup \{u\}) - H(A_i)\big). \end{aligned}$$

There are at most r terms in this sum (since $|A^\star| = r$). By construction, greedy selects the token with the largest marginal gain at each step, so each term is at most $H(A_{i+1}) - H(A_i)$. Hence

$$H(A^\star) - H(A_i) \leq r\big(H(A_{i+1}) - H(A_i)\big).$$

Rearranging:

$$H(A^\star) - H(A_{i+1}) \leq \Big(1 - \frac{1}{r}\Big)\big(H(A^\star) - H(A_i)\big).$$

This is a contraction: the gap to optimality shrinks by a factor of $(1 - 1/r)$ at each iteration. Iterating from $i = 0$ to $i = r - 1$ and using $H(A_0) = 0$:

$$H(A^\star) - H(A_r) \leq \Big(1 - \frac{1}{r}\Big)^r H(A^\star).$$

Thus $H(A_r) \geq \big(1 - (1 - 1/r)^r\big)H(A^\star)$. Finally, the inequality $(1 - 1/r)^r \leq e^{-1}$ (which holds for all $r \geq 1$) gives the $(1 - 1/e)$ bound. □

A.5 Proof of theorem 3.5 (First-order approximation and error bound)

Theorem (Restatement of theorem 3.5)**.** Assume $R_c(A) \leq \rho < 1$ for all feasible A and c. Then

$$F(A) = -\mathbb{E}[R_c(A)] - \mathbb{E}[\eta(R_c(A))],$$

where $\eta(r) := \sum_{k=2}^{\infty} r^k/k$ satisfies $0 \leq \eta(r) \leq r^2/(2(1-\rho))$. The first-order surrogate is frequency pruning.

The proof uses the Taylor expansion of $\log(1 - r)$ and bounds the remainder term.

Proof. For $r \in [0, 1)$, the Taylor expansion of the logarithm gives

$$\log(1-r) = -\sum_{k=1}^{\infty} \frac{r^k}{k} = -r - \underbrace{\sum_{k=2}^{\infty} \frac{r^k}{k}}_{=:\eta(r)}.$$

The remainder $\eta(r)$ is clearly nonnegative. To bound it from above for $r \in [0, \rho]$:

$$\begin{aligned}\eta(r) &= \sum_{k=2}^{\infty} \frac{r^k}{k} \le \frac{1}{2} \sum_{k=2}^{\infty} r^k \\ &= \frac{r^2}{2(1-r)} \le \frac{r^2}{2(1-\rho)},\end{aligned}$$

where the first inequality uses $1/k \le 1/2$ for $k \ge 2$, and the second uses $r \le \rho$.

Now substitute $r = R_c(A) = 1 - M_c(A)$ and take expectations:

$$\begin{aligned}F(A) &= \mathbb{E}[\log(1 - R_c(A))] \\ &= -\mathbb{E}[R_c(A)] - \mathbb{E}[\eta(R_c(A))].\end{aligned}$$

To identify the first-order term, expand $\mathbb{E}[R_c(A)]$:

$$\mathbb{E}[R_c(A)] = 1 - \mathbb{E}\Big[\sum_{v \in B} p_c(v)\Big] - \sum_{u \in A} m(u).$$

The first two terms are independent of A. Therefore, minimizing the first-order contribution $\mathbb{E}[R_c(A)]$ is equivalent to maximizing $\sum_{u \in A} m(u)$, which is exactly frequency pruning (theorem 3.2). □

A.6 Proof of theorem 3.6 (Common-ranking condition)

Theorem (Restatement of theorem 3.6)**.** If there exists a total order $u_1 \succ \cdots \succ u_{|U|}$ on U such that $p_c(u_1) \ge \cdots \ge p_c(u_{|U|})$ for every context c, then $A_{\text{rank}} := \{u_1, \ldots, u_r\}$ maximizes $F(A)$ over all $|A| = r$, and coincides with the top-r set by marginal frequency.

The proof shows that a common ranking makes the top-r tokens simultaneously optimal for every context, not just on average. This is a much stronger condition than what the greedy algorithm requires.

Proof. Fix any feasible set $A \subseteq U$ with $|A| = r$, and fix any context c. Because $u_1, \ldots, u_r$ are the r tokens with the largest probabilities under p_c (by the common-ranking assumption), we have

$$\sum_{j=1}^{r} p_c(u_j) \ge \sum_{u \in A} p_c(u).$$

Adding the baseline mass $\sum_{v \in B} p_c(v)$ to both sides gives

$$M_c(A_{\text{rank}}) \ge M_c(A).$$

Since log is an increasing function:

$$\log M_c(A_{\text{rank}}) \ge \log M_c(A).$$

This holds for *every* context c. Taking expectation over $c \sim \mu$:

$$F(A_{\text{rank}}) \ge F(A).$$

Since A was arbitrary, A_{rank} is optimal.

For the second claim: if $u_i \succ u_j$, then $p_c(u_i) \ge p_c(u_j)$ for every context c. Taking expectations gives $m(u_i) = \mathbb{E}[p_c(u_i)] \ge \mathbb{E}[p_c(u_j)] = m(u_j)$. Hence the common ranking is also the ranking by marginal predictive frequency, and A_{rank} is the frequency solution. □

A.7 Proof of theorem 3.7 (Frequency can be strictly suboptimal)

Theorem (Restatement of theorem 3.7)**.** There exist a deployment distribution over contexts and a budget r such that the support maximizing $\sum_{u \in A} m(u)$ is not a maximizer of $F(A)$.

This is a minimal counterexample with three optional tokens and two contexts. It demonstrates the generalist-vs-specialist tradeoff that is central to the paper: a globally frequent "generalist" token is preferred by the linear criterion, but two domain-specific "specialist" tokens yield higher F by avoiding catastrophic undercoverage.

Proof. Let the optional vocabulary be $U = \{u, b, c\}$ and let the budget be $r = 2$. Assume the fallback support contributes probability 0.01 in every context. Consider two contexts, c_1 and c_2, each occurring with probability $1/2$, with teacher probabilities:

$$p_{c_1}(u) = 0.34, \quad p_{c_1}(b) = 0.65, \quad p_{c_1}(c) = 0,$$
$$p_{c_2}(u) = 0.34, \quad p_{c_2}(b) = 0, \quad p_{c_2}(c) = 0.65.$$

The marginal predictive frequencies are $m(u) = 0.34$, $m(b) = 0.325$, $m(c) = 0.325$. Since $m(u) > m(b) = m(c)$, the frequency solution selects u and one of b, c; without loss of generality, $A_{\text{freq}} = \{u, b\}$.

Under A_{freq}, the retained masses are:

$$M_{c_1}(A_{\text{freq}}) = 0.01 + 0.34 + 0.65 = 1.00,$$
$$M_{c_2}(A_{\text{freq}}) = 0.01 + 0.34 + 0 = 0.35.$$

Context c_1 is perfectly covered, but context c_2 retains only 35% of its mass. The objective value is

$$\begin{aligned} F(A_{\text{freq}}) &= \tfrac{1}{2} \log 1.00 + \tfrac{1}{2} \log 0.35 \\ &= \tfrac{1}{2} \log 0.35 \approx -0.525. \end{aligned}$$

Now consider the alternative $A_{\text{spec}} = \{b, c\}$. Its retained masses are:

$$M_{c_1}(A_{\text{spec}}) = 0.01 + 0.65 + 0 = 0.66,$$
$$M_{c_2}(A_{\text{spec}}) = 0.01 + 0 + 0.65 = 0.66.$$

Both contexts are covered equally. The objective value is

$$F(A_{\text{spec}}) = \log 0.66 \approx -0.416.$$

Since $-0.416 > -0.525$, we have $F(A_{\text{spec}}) > F(A_{\text{freq}})$, and the frequency solution is strictly suboptimal.

The intuition is clear: token u is the "generalist": it has moderate mass in both contexts. Tokens b and c are "specialists": each is critical in exactly one context and useless in the other. The linear criterion prefers the generalist because it contributes more mass on average. But the log criterion recognizes that the generalist's contribution to c_1 (where coverage is already excellent) is worth much less than the specialist's contribution to c_2 (where coverage is catastrophically low). This is exactly the diminishing-returns mechanism formalized by theorem 3.4. □

B Lazy-Greedy Algorithm

Algorithm 1 below gives the lazy-greedy procedure used in all experiments. Submodularity of F guarantees that marginal gains are non-increasing as the support A grows, so any previously computed gain is a valid upper bound on its current value. Maintaining a max-heap keyed by these upper bounds and recomputing a candidate's gain only when it reaches the top of the heap produces the same selection as standard greedy while performing far fewer gain evaluations in practice.

Algorithm 1 Lazy Greedy for Log-Retained-Mass Maximization

Require: Calibration distributions $p_1, \ldots, p_n$ over V; budget r; fallback mass β
Ensure: Selected support A with $|A| = r$

$M_i \leftarrow \beta$ for all $i = 1, \ldots, n$
Compute $\Delta(v) \leftarrow \frac{1}{n} \sum_{i=1}^{n} \log(1 + p_i(v)/M_i)$ for all $v \in U$
Insert all $(v, \Delta(v))$ into max-heap $\mathcal{H}$
$A \leftarrow \varnothing$
for $k = 1$ **to** r **do**
  repeat
    Pop (v, key) from $\mathcal{H}$
    $\Delta \leftarrow \frac{1}{n} \sum_{i=1}^{n} \log(1 + p_i(v)/M_i)$ {recompute exact gain}
    if $\Delta \geq$ top key of $\mathcal{H}$ **then**
      $A \leftarrow A \cup \{v\}$; $M_i \leftarrow M_i + p_i(v)$ for all i
    else
      Push (v, Δ) back into $\mathcal{H}$
    end if
  until $|A| = k$
end for
return A

Proposition B.1 (Lazy greedy is exact)**.** *Let u be the candidate accepted at termination of a given round of algorithm 1, and let $U(w)$ denote the stored upper bound on each remaining candidate $w \in U \setminus A$. Then u has the largest exact current marginal gain among all remaining candidates, and is precisely the token that standard greedy would select.*

Proof. By the stopping condition, after recomputation the exact gain of u satisfies $\Delta(u \mid A) \geq U(w)$ for every remaining candidate w. Since each stored key $U(w)$ is an upper bound on $\Delta(w \mid A)$ (submodularity: marginal gains cannot increase as A grows), $\Delta(w \mid A) \leq U(w) \leq \Delta(u \mid A)$ for all w. Hence u is the exact argmax of $\Delta(\cdot \mid A)$, which is what standard greedy selects. □

C Experimental Details

C.1 Synthetic multi-domain experiment

Vocabulary structure. The vocabulary of $|V|$=10,000 tokens is partitioned as follows: a *shared block* of 500 tokens (indices 0–499) common to all domains; up to 10 *domain-specific blocks* of 500 tokens each (domain d occupies indices $500{+}500d$ through $999{+}500d$); and a *tail block* of 4,500 tokens (indices 5,500–9,999). Within each domain block, 50 tokens are designated as a *hot subset*, selected once using a fixed random seed (seed = 42) and shared across all configurations.

Domain prototypes. For each active domain d, we define a prototype next-token distribution $\pi^{(d)} \in \Delta(V)$ that places: mass 0.55 uniformly on its 50 hot tokens (0.011 per token); mass 0.30 uniformly on the 500 shared tokens (0.0006 per token); mass 0.10 uniformly on the remaining 450 tokens in its own block ($\sim$0.00022 per token); and mass 0.05 uniformly on all other tokens ($\sim 5.6 \times 10^{-6}$ per token).

Context sampling. Individual contexts are sampled as draws from a Dirichlet distribution centered on the prototype: $p_i \sim \text{Dirichlet}(\alpha \cdot \pi^{(d)})$, where $\alpha = 80$ is the concentration parameter. Higher α produces contexts closer to the prototype; lower α increases within-domain variation.

Domain mixing. For D active domains, domain $d \in \{0, \ldots, D{-}1\}$ receives weight $w_d \propto 1/(d{+}1)$ (harmonic weights). The number of calibration contexts from domain d is $\lfloor w_d \cdot$

5,000⌉, and similarly for the 2,500 evaluation contexts. Calibration and evaluation sets use different random seeds (calibration: 1000+D; evaluation: 2000+D) but the same hot-subset selection.

Support sizes. Experiment A (support sweep): D=10, $r \in \{100, 300, 500, 700, 900\}$. Experiment B (domain sweep): r=50, $D \in \{1, 2, \ldots, 10\}$. The same calibration/evaluation data is reused across all r values in Experiment A.

C.2 Multilingual distributional experiment

Model. Qwen3.5-0.8B-Base (Qwen Team, 2026), a 0.8B-parameter autoregressive language model with vocabulary size 248,320 and native multilingual support.

Languages. We select 10 typologically diverse languages from FLORES-200 (NLLB Team, 2022): English (en), French (fr), Spanish (es), German (de), Swahili (sw) [Latin script]; Russian (ru) [Cyrillic]; Arabic (ar) [Arabic script]; Hindi (hi) [Devanagari]; Japanese (ja), Chinese (zh) [CJK].

Context construction. For each language, we tokenize all sentences in the FLORES-200 `dev` split (calibration) and `devtest` split (evaluation). From each sentence, we extract contexts at every token position, yielding ∼1,000 contexts per language per split. For each context, we store the top-8,192 next-token probabilities as a sparse representation of the teacher distribution.

Calibration regimes. *Uniform:* all languages contribute equally ($w_\ell = 1/10$). *English-dominant:* English receives $w_{\text{en}} = 0.6$; the remaining 9 languages share 0.4 equally ($w_\ell \approx 0.044$). Evaluation is always uniform across all languages.

Support sizes. $K \in \{1{,}000,\ 2{,}000,\ 4{,}000,\ 6{,}000,\ 10{,}000\}$.

C.3 Multilingual text continuation

Generation protocol. For each of the ∼1,000 evaluation sentences per language, we take the first 60% of tokens as a prompt and generate a continuation of up to 80 tokens using greedy (argmax) decoding. At each decoding step, the model scores only the tokens in the selected support A; all other tokens receive $-\infty$ logits.

Evaluation metric. We use chrF (Popović, 2015) (character n-gram F-score, $n \leq 6$), which handles all scripts uniformly without requiring word segmentation. We report per-language chrF and the macro average across the six evaluation languages (en, fr, ar, ja, ru, zh).

Calibration. English-dominant weights (60% English, 40% split among the other nine calibration languages, matching the distributional experiment). The same support sets from the distributional experiment are reused.

C.4 Robustness across deployment types

Deployments. The robustness study (section 4.4) evaluates four deployments at a fixed budget K=1,000, each defined by a set of labels and a calibration weight on the dominant label (the remainder split equally among the others): *Latin (5 langs)*, five Latin-script languages from Wikipedia (en 0.60; fr/de/es/it 0.10 each); *Multi-script (10 langs)*, the ten FLORES-200 languages with graduated weights (en 0.60; fr/es 0.08; de 0.05; ru/ar 0.04; zh/hi/ja 0.03; sw 0.02), spanning five scripts; *Multi-script (5 langs)*, five disjoint scripts from Wikipedia (Latin/en 0.60; Cyrillic/ru, Arabic/ar, CJK/zh, Devanagari/hi 0.10 each); and *High skew (en+ar, 0.99)*, two FLORES-200 languages at extreme skew (en 0.99; ar 0.01). Calibration weights the contexts of each label by its share; evaluation is uniform over labels. Context construction, the top-8,192 sparse teacher representation, and the chrF generation protocol are identical to Appendices C.2 and C.3.

Magnitude baseline. In addition to frequency and greedy, the robustness study includes *magnitude*, a data-free baseline that applies the magnitude-based saliency criterion of Han et al. (2015) at the granularity of output-token rows, keeping the K tokens with the largest L_2 norm of their `lm_head` row. It uses no calibration data and therefore serves as a lower bound that isolates the value of calibrating the support to the deployment distribution. All pruned methods share the same fallback support and keep the same number of tokens, so differences reflect *which* tokens each method retains rather than how many.

D Additional empirical results

D.1 Cross-family validation on Gemma-3-1B

To verify that the qualitative pattern of section 4.2 is not specific to the Qwen tokenizer family, we re-run the multilingual distributional experiment on Gemma-3-1B-PT (Gemma Team, 2025) (V=262,144), using the same FLORES-10 corpus and the same graduated English-dominant calibration weights. Table 3 reports global reverse-KL at three support budgets and table 4 reports the per-language breakdown at K=10,000.

Method	K=4,000	K=10,000	K=20,000
Magnitude	4.02	2.96	2.55
Frequency	4.05	0.76	0.354
Greedy	**4.05**	**0.73**	**0.348**

Table 3: Global reverse KL on Gemma-3-1B-PT, FLORES-10, English-dominant calibration. At K=4,000 all three methods sit near the β-floor because Gemma-3's 262k-token vocabulary is too sparse for any small budget to cover the per-context tops. Once K admits those tops, greedy is consistently ahead of frequency, with the gap of ~ 0.03 nats at K=10,000 shrinking to ~ 0.006 at K=20,000 as both approach the full-vocab ceiling. The data-free magnitude baseline is much larger than the data-driven methods once they separate from the floor.

Method	en	fr	es	de	ru	ar	zh	hi	ja	sw
Frequency	**0.28**	0.62	0.58	**0.77**	1.21	0.75	0.97	0.84	1.02	**0.58**
Greedy	0.32	**0.64**	**0.59**	**0.77**	**1.12**	**0.73**	**0.89**	**0.74**	**0.93**	0.58

Table 4: Per-language reverse KL on Gemma-3-1B-PT at K=10,000. Greedy improves Russian, Chinese, Hindi, and Japanese (-0.08 to -0.10 nats each) at a cost of $+0.04$ on English, mirroring the Qwen3.5-0.8B pattern of fig. 2 (right). Frequency is slightly better on three Latin-script languages; greedy is strictly better on every non-Latin script.

D.2 Calibration-skew sweep on FLORES-10

Section 4.2 reports two calibration regimes (uniform and 60% English). To map the dependence of the greedy advantage on calibration skew, we re-run the FLORES-10 cell at K=1,000 with the dominant English weight varied across $\{0.50, 0.75, 0.90, 0.99\}$ (remainder split equally among the other nine languages); table 5 reports global reverse-KL for each method.

D.3 Speculative-decoding: full-matrix aggregate and γ sweeps

The body table 2 reports per-language results for the two deployments where the underlying reverse-KL gap is non-trivial. Table 6 gives the full aggregate-acceptance matrix on all four deployments, with the magnitude data-free baseline included.

Draft length sensitivity. We repeat the high-skew cell at draft lengths $\gamma \in \{2, 4, 8\}$ (table 7). Greedy's per-token acceptance is essentially constant in γ (~ 0.40), while frequency drifts

en weight	Magnitude	Frequency	Greedy	Δ (freq−greedy)
0.50	3.78	1.57	**1.44**	+0.13
0.75	3.78	1.83	**1.57**	+0.26
0.90	3.78	2.18	**1.84**	+0.35
0.99	3.78	2.92	**2.51**	+0.41

Table 5: Global reverse KL on FLORES-10 at K=1,000 as a function of calibration skew (Qwen3.5-0.8B-Base). The greedy−frequency gap grows monotonically with skew, in line with theorem 3.7. Magnitude is independent of calibration weights, so its value is constant across rows.

Deployment	Magn.	Freq.	Greedy	Δ pp, 95% CI
Latin (5 langs)	0.08	**0.49**	0.48	−0.9 [−3.0, +1.3]
Multi-script (10 langs)	0.05	0.37	**0.38**	+1.4 [+0.4, +2.5]
Multi-script (5 langs)	0.05	**0.44**	0.43	−0.7 [−2.2, +0.9]
High skew (en+ar, 0.99)	0.05	0.37	**0.40**	+3.3 [+1.6, +4.9]

Table 6: Aggregate sample-based speculative-decoding draft acceptance on all four deployments of table 1 at K=1,000 (Qwen3.5-0.8B-Base). Magnitude is the data-free baseline; bold marks the best pruned method. The two deployments with the smallest reverse-KL gaps in table 1 (Latin and Multi-script-5, both $\leq$ 0.08 nats) have $|\Delta| \leq$ 0.9pp with CIs straddling zero, consistent with the theoretical prediction that small reverse-KL gaps yield small acceptance gaps.

slightly downward as γ grows, so the paired gap widens from +3.4pp at γ=2 to +4.3pp at γ=8. The qualitative finding does not depend on this hyper-parameter, and longer draft chains slightly favor greedy because the chance of crossing a specialist-token boundary that frequency mis-handles grows with γ.

γ	Frequency	Greedy	Δ pp, 95% CI
2	0.37	**0.41**	+3.4 [+1.9, +4.8]
4	0.37	**0.40**	+3.3 [+1.6, +4.9]
8	0.36	**0.40**	+4.3 [+2.6, +5.8]

Table 7: Acceptance on the high-skew deployment (en+ar, 0.99) at three draft lengths γ. Greedy's gain is robust across γ and grows slightly with longer drafts. Δ is the per-prompt stratified paired-bootstrap mean difference (greedy − frequency), n=80 prompts.