VOCABULARY PRUNING VIA SUBMODULAR OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models generate text by scoring a distribution over the full vocabulary at each decoding step, making vocabulary size a significant latency and memory component, particularly in speculative decoding, early-exit inference, and edge deployment. Vocabulary pruning addresses this by retaining only a fixed subset of K tokens, but existing approaches rely on frequency-based heuristics that optimize average token coverage and can leave minority languages or specialized domains catastrophically uncovered. Here, we formalize vocabulary pruning as a combinatorial optimization problem and propose maximizing the expected log retained probability mass, an objective equivalent to minimizing the expected reverse KL divergence to the teacher distribution. The objective is monotone submodular, yielding a greedy algorithm with a $(1 - 1/e)$-approximation guarantee. Frequency pruning is its first-order approximation: it is optimal under a sufficient common-ranking condition and can be suboptimal when the condition fails. Our experiments show that vocabulary pruning reduces memory and latency, while the proposed support selection improves accepted length and throughput in speculative decoding at the same vocabulary budget.

1 INTRODUCTION

Large language models (LLMs) have become a default interface for reasoning, coding, search, and interactive assistance. As these models continue to scale, inference has emerged as the central systems bottleneck: deployment must pay the cost of autoregressive generation token by token, often under tight latency, memory, or energy constraints. This tension has motivated a large body of work on accelerating inference through algorithmic changes, including speculative decoding (Chen et al., 2023; Leviathan et al., 2023; Li et al., 2024b; Zhao et al., 2025), early-exit mechanisms (Schuster et al., 2022; Elhoushi et al., 2024; Vincenti et al., 2024), and post-hoc compression of model components.

A common theme behind these efforts is that not all parts of the computation are equally useful at every decoding step. In both speculative decoding and early-exit systems, the *vocabulary-sized* part of the computation can become a bottleneck: the model repeatedly maps hidden states to a distribution over tens or hundreds of thousands of output tokens, 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).

Vocabulary pruning offers one route to faster inference. 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). There are two main strategies. *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. In this paper, we start by theoretically studying the output-side pruning:

> *If one can afford to score only K output tokens, which K tokens should be kept so that the behavior of the original model is preserved as much as possible?*

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.

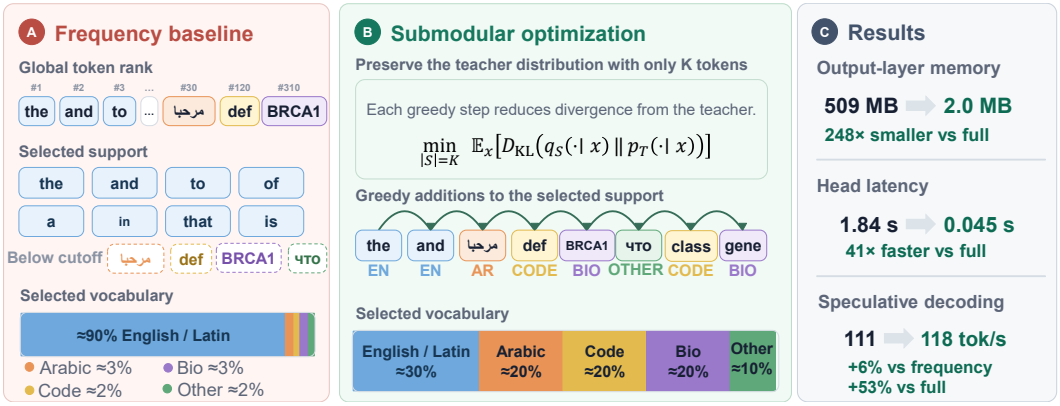


Figure 1: **The same small vocabulary can cover deployment contexts very differently. (A)** Selecting the most frequent tokens favors the popular domain and can exclude tokens that matter elsewhere. **(B)** Our method repeatedly chooses the token that helps the least-covered contexts most, producing a more balanced set of K output tokens. **(C)** The smaller output layer reduces memory and latency, while the improved token set makes speculative decoding faster than frequency approach.

This question has been studied in speculative decoding, where a reduced LM head directly 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). Particularly, in speculative decoding output-vocabulary pruning preserves the exactness of the original method while reducing the draft model's latency and memory footprint.

One common approach for vocabulary pruning is based on frequency of tokens in calibration corpus. As such this approach is simple, interpretable, and often effective. 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 this output-side problem. We fix a pretrained teacher and a deployment distribution over contexts, leaving the model and tokenizer unchanged and compressing only the output support. Restricting the output space projects 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 objective is exactly equivalent to minimizing the expected reverse KL divergence between the teacher distribution and its support-restricted renormalization. The logarithm penalizes supports that leave some contexts badly under-covered and induces a monotone submodular set function, yielding a near-optimal greedy algorithm with a formal $(1 - 1/e)$-approximation guarantee. Figure 1 gives an intuitive view of the problem and the resulting selection rule.

We show that frequency-based selection optimizes a linearized objective. It is optimal under a common-ranking condition across contexts and can become suboptimal when contexts disagree on which tokens matter. This helps explain why simple static pruning rules can offer substantial speedups yet remain brittle under shifts across domains, languages, or generation modes (Bogoychev et al., 2024; Zhao et al., 2025; Vincenti et al., 2024).

This paper makes two contributions:

- We formulate output-vocabulary pruning as deployment-aware support selection. Our log-retained-mass objective is equivalent to minimizing expected reverse KL under support restriction and is monotone submodular, enabling $(1-1/e)$-approximate greedy selection with exact lazy evaluation.
- We characterize frequency pruning as a first-order approximation, establish when it is optimal, and show when it fails. Across real and synthetic datasets, our experiments compare greedy with frequency and magnitude at equal budgets, as well as in applications such as speculative decoding.

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$\times$ 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 $1.24\times$ 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: we show that the frequency ranking used by FR-Spec is the first-order approximation to a principled submodular objective, and identify precisely when and why it is suboptimal.

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; Liang et al., 2023; Bogoychev et al., 2024; Dorkin et al., 2025). More general approaches include vocabulary compression 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-support pruning* 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? Because only the support of the next-token distribution changes, the theory is clean. We introduce a log-retained-mass objective and show that it has an exact reverse-KL interpretation (Theorem 3.3), is monotone submodular (Theorem 3.4), admits a greedy $(1 - 1/e)$ approximation guarantee (Theorem 3.5), recovers frequency pruning as a first-order surrogate (Theorem 3.6), and can strictly outperform frequency in heterogeneous deployments (Theorem 3.8). All proofs are deferred to 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. For the population objective, also assume $\mathbb{E}_\mu[-\log M_c(\varnothing)] < \infty$; this is automatic for a finite calibration set with positive fallback mass. 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

$$A^\star \in \operatorname*{argmax}_{\substack{A \subseteq U \\ |A|=r}} F(A), \quad F(A) := \mathbb{E}_{c\sim\mu}\big[\log M_c(A)\big]. \tag{2}$$

The reverse-KL interpretation in Section 3.3 motivates the log objective. We first examine the corresponding linear alternative.

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 and offers no protection against a small set of contexts being covered very poorly. Our objective replaces this linear criterion with a concave one.

3.3 FROM SUPPORT RESTRICTION TO A REVERSE-KL OBJECTIVE

For a fixed context c and support A, consider the family of distributions supported on the retained set $\mathcal{Q}_A := \{q \in \Delta(V) : \operatorname{supp}(q) \subseteq S_A\}$, where $\Delta(V)$ is the probability simplex on V. The natural reduced distribution is obtained by restricting the teacher to S_A and renormalizing:

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

The next theorem identifies this construction as the reverse-KL projection onto the restricted support. (Forward KL, $D_{\mathrm{KL}}(p_c\|q)$, is infinite whenever $\operatorname{supp}(q) \subset \operatorname{supp}(p_c)$, which is the entire purpose of pruning; reverse KL is the uniquely finite KL-based criterion for support selection.)

Theorem 3.3 (Exact reverse-KL characterization)**.** *Fix a context c and a feasible support A. Then $q^*_{c,A}$ defined in Equation* (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 main idea of the proof is to decompose $D_{\mathrm{KL}}(q\|p_c)$ into $D_{\mathrm{KL}}(q\|q^*_{c,A}) - \log M_c(A)$, so the minimum over q is achieved when the first term vanishes. The full proof is in Appendix A.

The induced global objective. Theorem 3.3 leads directly to the population objective in Definition 3.1:

$$\begin{aligned} F(A) &:= \mathbb{E}_{c\sim\mu}[\log M_c(A)], \\ J(A) &:= -F(A) = \mathbb{E}_{c\sim\mu}[-\log M_c(A)]. \end{aligned} \tag{4}$$

Maximizing F is equivalent to minimizing the expected reverse-KL distortion between the teacher distribution and its best support-restricted renormalization. The objective also maximizes the weighted geometric mean of retained masses, connecting it to Nash social welfare and proportional-fairness objectives. We use this as motivation, without claiming uniqueness from scale invariance and Pareto efficiency alone.

3.4 THE STRUCTURAL RESULT: MONOTONE SUBMODULARITY

The main technical advantage of the objective in Equation (4) is that it defines a monotone submodular set function. For each fixed context c, define $f_c(A) := \log M_c(A)$, so that $F(A) = \mathbb{E}[f_c(A)]$.

Theorem 3.4 (Monotonicity and submodularity)**.** *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 (in Appendix A) follows from the explicit form of the marginal gain. For a candidate token u, the one-context gain is

$$\Delta_c(u \mid A) := f_c(A \cup \{u\}) - f_c(A) = \log\Big(1 + \frac{p_c(u)}{M_c(A)}\Big). \tag{5}$$

This is nonnegative (monotonicity) and decreasing in $M_c(A)$ (submodularity): as the current support grows, the denominator increases and the marginal gain shrinks. This is the diminishing-returns property.

A token's value depends on its probability mass relative to the current coverage of the context. The log objective gives more weight to poorly covered contexts, while frequency pruning treats every unit of mass equally.

3.5 Greedy algorithm and approximation guarantees

Because F is monotone submodular, the cardinality-constrained problem $\max_{|A|=r} F(A)$ can be approximated by greedy maximization. Starting from $A_0 = \varnothing$, greedy constructs A_{i+1} from A_i by selecting

$$u_i \in \operatorname*{argmax}_{u \in U \setminus A_i} \big(F(A_i \cup \{u\}) - F(A_i)\big)$$

and setting $A_{i+1} = A_i \cup \{u_i\}$.

Theorem 3.5 (Hardness and greedy approximation)**.**

(i) Maximizing a monotone submodular function subject to a cardinality constraint is NP-hard.
(ii) No polynomial-time algorithm can achieve a ratio better than $(1-1/e)$ unless P=NP.
(iii) Greedy achieves this bound: for optimal $A^\star$,

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

Parts (i)–(ii) are classical (Feige, 1998); part (iii) is the guarantee of Nemhauser et al. (1978), with the specific proof in Appendix A. Greedy consequently has a formal near-optimality guarantee that, for the general monotone submodular class, no polynomial-time algorithm can exceed; we do not claim a matching lower bound for the specific log-retained-mass family.

From population to empirical objective. In practice, the deployment distribution μ is unknown. We work with a calibration set of contexts $c_1, \ldots, c_n \sim \mu$. The empirical counterpart of Equation (4) is

$$\widehat{F}(A) := \tfrac{1}{n} \textstyle\sum_{i=1}^{n} \log M_i(A), \tag{6}$$

where $M_i(A) := \sum_{v \in B} p_i(v) + \sum_{u \in A} p_i(u)$ and $p_i(v) := p_{c_i}(v)$. The exact marginal gain of a candidate token $u \notin A$ is

$$\Delta(u \mid A) = \frac{1}{n} \sum_{i=1}^{n} \log\Big(1 + \frac{p_i(u)}{M_i(A)}\Big). \tag{7}$$

Lazy greedy. The diminishing-returns property of submodularity enables an exact acceleration: the lazy-greedy variant maintains a max-heap of upper bounds on marginal gains and recomputes only when a candidate is popped. Because gains cannot increase as the support grows, any previously computed gain is a valid upper bound, so lazy greedy returns the identical solution as standard greedy at a fraction of the cost. The full procedure (Algorithm 1) and exactness proof are in Appendix B.

3.6 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 3.6 (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 (Proposition 3.2).

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

Under a strong structural condition, frequency gives the exact solution.

Theorem 3.7 (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 $A_{\mathrm{rank}} := \{u_1, \ldots, u_r\}$ maximizes $F(A)$ over all $|A| = r$, and coincides with the top-r set by marginal frequency.*

Frequency pruning is optimal when all contexts agree on the same token ranking. Heterogeneous deployments may violate this common-ranking assumption. When contexts disagree on which tokens are important, as often occurs across code, natural language, and specialized domains, frequency can fail.

The following construction gives a concrete example:

Theorem 3.8 (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. Token u is slightly more frequent globally, so the linear criterion keeps it. But tokens b and c each rescue a different context. The log objective values that diversity of coverage more than the extra average mass contributed by u. This example motivates testing whether the log-based selection rule improves minority-context coverage in heterogeneous deployments. It does not imply that frequency always ignores minority tokens or that greedy improves every context.

Scope. The theory concerns a fixed global support chosen offline; per-context online selection is trivial when the full teacher distribution is available. It covers output-side support restriction with a frozen tokenizer and hidden states, reducing the cost of scoring the output layer from $O(|V|d)$ to $O((|B| + r)d)$ without claiming any tokenizer-level compression. The fallback support B is part of the design space: any always-available support (e.g., a byte-level fallback or a small mandatory shortlist) that keeps the reverse-KL objective finite.

4 Experiments

We compare support selection across model families and deployment mixtures, then measure output-layer efficiency and end-to-end speculative decoding. In the clean quality tables, K counts all retained tokens, including mandatory tokens, and defaults to $1{,}000$. Distributional quality uses the smoothed score $\widehat{J}_\beta$ defined in Appendix C.2, rather than exact KL. The earlier controlled and multilingual figures provide separate exploratory analyses.

4.1 Quality across model families

We begin with the broadest test: the same ten-language, multi-script evaluation is repeated on four models spanning 0.5B to 7B parameters and vocabularies from 152k to 256k tokens. We compare data-free magnitude pruning, frequency pruning, and greedy log-mass selection. $\widehat{J}_\beta$ measures preservation of the teacher distribution, while chrF measures downstream continuation quality.

Table 1 shows a consistent ordering across model scales and vocabulary sizes. Greedy has lower $\widehat{J}_\beta$ than frequency for every model, and the same selected supports also yield higher chrF. Magnitude pruning is substantially worse, showing that output-row norm is not a sufficient proxy for deployment relevance. The agreement of the two metrics is important: the objective improves teacher-distribution fidelity, and that improvement survives autoregressive generation.

	$\widehat{J}_\beta \downarrow$			chrF ↑			
Model	Mag.	Freq.	Greedy	Full	Mag.	Freq.	Greedy
Qwen2.5-0.5B	4.592	1.472	**1.279**	12.52	1.04	7.92	**8.41**
Qwen3.5-0.8B	3.785	1.710	**1.525**	12.07	1.50	5.94	**6.63**
Qwen2.5-7B	4.600	1.707	**1.518**	15.94	1.00	10.77	**11.47**
Gemma-2-2B	4.064	1.317	**1.208**	4.06	0.21	1.92	**1.98**

Table 1: Cross-model quality on the ten-language multi-script deployment at $K{=}1{,}000$. Greedy improves both $\widehat{J}_\beta$ and chrF over frequency and magnitude for every model.

4.2 Deployment heterogeneity

We next test the condition highlighted by the theory. Using Qwen3.5-0.8B-Base, we compare five Latin-script languages, ten languages spanning five scripts, five languages with disjoint scripts, and a highly skewed English and Arabic mixture. Calibration follows the deployment weights (Appendix C.5); evaluation reports macro $\widehat{J}_\beta$ and chrF over the included languages. Calibration weights the contexts of each label by its share, while evaluation is uniform over labels.

	$\widehat{J}_\beta \downarrow$			chrF ↑			
Deployment	Mag.	Freq.	Greedy	Full	Mag.	Freq.	Greedy
Latin (5 langs)	3.50	1.22	**1.21**	18.6	2.5	10.8	**11.9**
Multi-script (10)	3.78	1.71	**1.53**	12.1	1.5	5.9	**6.6**
Multi-script (5)	3.84	1.32	**1.25**	14.0	0.5	7.3	**7.7**
High skew (en+ar, .99)	3.86	1.95	**1.55**	13.9	1.2	6.9	**8.9**

Table 2: Robustness across deployment types on Qwen3.5-0.8B-Base at $K{=}1{,}000$. Bold marks the best pruned method. Greedy is nearly tied with frequency in the low-heterogeneity Latin setting and improves as token rankings diverge across scripts or under extreme calibration skew.

Table 2 matches the common-ranking analysis. Frequency and greedy are almost tied in $\widehat{J}_\beta$ on the Latin deployment, where token rankings are comparatively aligned. The gap grows on multi-script mixtures and is largest under 99% English calibration: greedy lowers $\widehat{J}_\beta$ by 0.40 nats and improves chrF by 2.0 points. The result is therefore not a uniform penalty against frequency. It identifies the deployment regimes in which average token mass hides specialist tokens needed by minority contexts. All pruned methods share the same fallback support and retain the same number of tokens, so differences reflect which tokens each method keeps rather than how many.

4.3 Output-layer efficiency

The first two tables compare which tokens should be retained. We now separate that selection question from the generic systems benefit of retaining only K rows. At a fixed K, all selection rules use the same matrix shape. Table 3 reports the memory footprint and isolated projection-plus-softmax latency for 1,000 decoding steps.

	Output-layer memory (MB)			Latency / 1k steps (s)		
Model	Full	$K{=}$1k	Red.	Full	$K{=}$1k	Speedup
Qwen2.5-0.5B	272	1.8	152×	0.97	0.045	21×
Qwen3.5-0.8B	509	2.0	248×	1.84	0.045	41×
Qwen2.5-7B	1090	7.2	152×	3.82	0.048	80×
Gemma-2-2B	1180	4.6	256×	4.30	0.046	92×

Table 3: Generic output-layer benefit of pruning to $K{=}1{,}000$. Since the retained matrix shape is fixed, these savings apply equally to every selection rule.

The largest-vocabulary model reduces its output layer from 1,180 MB to 4.6 MB and its isolated latency from 4.30 s to 0.046 s. These savings motivate static pruning, but they do not by themselves

distinguish greedy from frequency. The distinction is quality at a fixed head size, as established in Tables 1 and 2. The isolated output-layer benchmark measures the component directly and should not be interpreted as an equal end-to-end model speedup.

4.4 END-TO-END SPECULATIVE DECODING

Finally, we integrate the selected support into FR-Spec/EAGLE-2 (Zhao et al., 2025; Li et al., 2024b). In the end-to-end FR-Spec experiments, the EAGLE-2 drafter retains full input embeddings and uses a separate, untied output matrix; physically removing unselected output rows reduces the memory and projection cost of that draft head while preserving full-vocabulary inputs. Full-vocabulary drafting offers the highest acceptance but pays for the wide LM head at every draft step. Frequency and greedy use the same $K{=}1{,}000$ rows, so their difference isolates support quality.

	Accepted length			Throughput (tok/s)		
Drafter model	Full	Freq.	Greedy	Full	Freq.	Greedy
Llama-3.2-1B	1.363	1.129	**1.199**	77	111	**118**
Qwen2-7B	1.491	1.265	**1.306**	25	29	**30**

Table 4: End-to-end FR-Spec/EAGLE-2 speculative decoding at $K{=}1{,}000$ on Arabic evaluation. Frequency and greedy use identical drafter-head sizes and differ only in selected tokens.

Table 4 demonstrates that pruning increases throughput relative to full-vocabulary drafting, while greedy recovers more accepted length than frequency at the same cost. On both model families this higher acceptance also yields higher measured throughput. On Llama-3.2-1B, greedy improves throughput by 6% over frequency and 52% over full-vocabulary drafting; on Qwen2-7B the corresponding improvements are 3% and 17%. With the Llama-3.2-1B EAGLE-2 drafter at $K{=}16$k, pruning yields large gains over full-vocabulary drafting on both English (144.1 to 211.3 tok/s) and multilingual text (94.0 to 127.9 tok/s); at this large budget the English-derived FR-Spec index and a deployment-matched vocabulary perform within one percent of each other, so the choice of selection rule matters at the smaller budgets studied above. In a separate paired run ($n{=}40$ prompts per language) at $K{=}1$k, greedy raises accepted length on every reported Llama language: Arabic $1.139 \to 1.233$, Russian $1.372 \to 1.402$, Chinese $1.159 \to 1.185$, Japanese $1.277 \to 1.295$, and Hindi $1.543 \to 1.558$; all five one-sided paired Wilcoxon tests on that run pass Bonferroni correction at $\alpha/5{=}0.01$ (per-prompt counts and caveats in Table 9). Complete per-language tables, the K-sweep, and a complementary sample-based acceptance study appear in Appendices E and F. Together, the four tables separate two complementary benefits: pruning reduces output-layer memory and latency, and submodular support selection preserves more useful draft tokens within the resulting budget.

4.5 CONTROLLED AND MULTILINGUAL ANALYSES

We next isolate the mechanism behind the four deployment-level results. The controlled experiment uses a vocabulary of 10,000 tokens with 500 shared tokens, up to ten domain-specific blocks, and 50 high-probability tokens per domain. Contexts are Dirichlet draws with concentration 80 around domain prototypes, and domains follow harmonic mixture weights; each configuration contains 5,000 calibration and 2,500 evaluation contexts. Figure 2 varies either the retained support or the number of active domains. The two rules coincide with one domain, but greedy separates as domains disagree and the budget tightens. In the $D{=}10$, $K{=}100$ cell, reverse KL falls from 2.53 for frequency to 1.84 for greedy, while the share of contexts left at the fallback floor drops from 49% to 31%.

Under English-dominant calibration, Figure 3 shows that the gap persists across support sizes and is concentrated on languages whose token rankings differ most from English. At $K{=}1{,}000$, greedy reduces reverse KL for Arabic, Japanese, Russian, and Chinese while giving up only 0.08 nats on English. For the continuation study in Figure 4, the first 60% of each sentence is used as a prompt, up to 80 tokens are generated with the selected support, and chrF is macro-averaged across scripts.

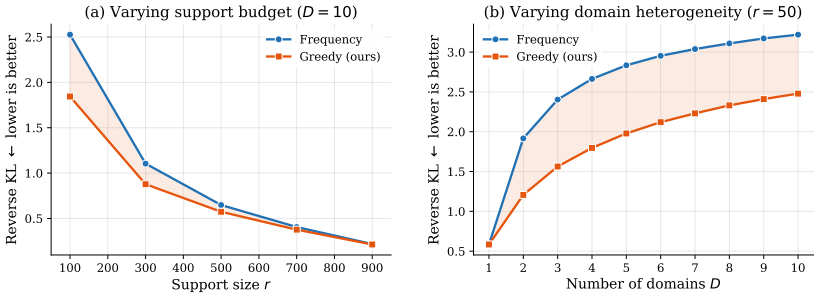
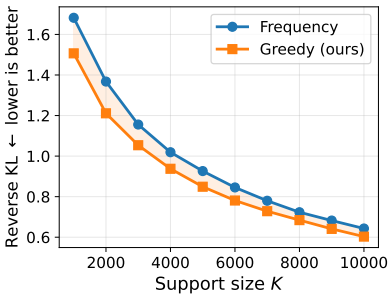


Figure 2: Controlled multi-domain experiment. Left: support-budget sweep with ten domains. Right: domain-heterogeneity sweep with $K=50$. Greedy improves most under tight budgets and increasing disagreement among domain-specific token rankings, while both methods coincide in the single-domain condition.



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 3: Multilingual distributional fidelity. Left: reverse KL versus support size under English-dominant calibration. Right: selected per-language values at $K=1{,}000$. Greedy gives up little English coverage while substantially improving languages with distinct scripts.

Fallback and evaluation scope. The mandatory-token set and the added smoothing floor are distinct (Appendix C.2). Neither shared smoothing nor stable selected supports would establish invariance of exact-KL rankings. The saved experiments support comparisons under the stated empirical score and evaluation weights.

5 Conclusion

In this paper, we studied how to select output vocabulary that preserves the behavior of a pretrained language model across heterogeneous deployment contexts. We formulated vocabulary pruning as maximizing expected log retained probability mass and showed that this objective has an exact reverse-KL interpretation, is monotone submodular, and can be optimized efficiently with a greedy approximation algorithm. We also characterized frequency pruning as a linear surrogate that works well when contexts agree on token importance but can neglect specialist tokens when languages, scripts, or domains have different token rankings. Our experiments confirmed this distinction across controlled mixtures, multilingual generation, and several model families: the proposed method remained close to frequency in homogeneous settings and produced more consistent improvements as deployment heterogeneity increased. We further evaluated the selected vocabularies in speculative decoding, where reducing the drafter's output head lowered its computational cost while the target model retained the full vocabulary. At the same vocabulary size, our method preserved more useful draft tokens than frequency pruning, resulting in greater accepted length and higher end-to-end throughput. Future work could extend this framework to dynamically selected vocabularies, online recalibration under distribution shift, hardware-aware token budgets, and broader deployment settings such as code, dialogue, and specialized domains.

AI USE STATEMENT

Generative AI tools assisted with editing, literature search, experiment tooling, and checks of mathematical claims and numerical results. The authors are responsible for the final text, analyses, and experimental evidence.

REPRODUCIBILITY STATEMENT

The theory and exact lazy-greedy procedure are given in Section 3 and appendices A and B. The clean empirical score and support conventions are defined separately in Appendix C.2. Saved per-prompt acceptance records and the corrected FR-Spec rerun support the reported aggregate checks. Limitations of the original figure-generation records and their separation from the clean result tables are documented in Appendix C.2. Data and code will be released publicly after the review decision.

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, Chris Lott, and Mingu Lee. 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.

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.

Davis Liang, Hila Gonen, Yuning Mao, Rui Hou, Naman Goyal, Marjan Ghazvininejad, Luke Zettlemoyer, and Madian Khabsa. XLM-V: Overcoming the vocabulary bottleneck in multilingual masked language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13142–13152, 2023.

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 PROPOSITION 3.2 (LINEAR RETAINED MASS YIELDS FREQUENCY PRUNING)

Proposition (Restatement of Proposition 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 (Equation (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:

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

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 Equation (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 $v \in S_A$, Equation (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

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

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 the diminishing-returns condition for 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 Proof of Theorem 3.5 (Hardness and greedy approximation)

Theorem (Restatement of Theorem 3.5)**.**

(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:

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

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 3.6 (First-order approximation and error bound)

Theorem (Restatement of Theorem 3.6)**.** 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 $r \in [0, \rho]$:

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

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:

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

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 (Proposition 3.2). □

A.6 Proof of Theorem 3.7 (Common-ranking condition)

Theorem (Restatement of Theorem 3.7)**.** 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. This condition is stronger than the requirements of the greedy algorithm.

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 3.8 (Frequency can be strictly suboptimal)

Theorem (Restatement of Theorem 3.8)**.** 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:

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

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

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

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

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

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.

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. The log criterion assigns less value to its contribution in the well-covered context c_1 than to the specialist's contribution in the poorly covered context c_2. This is the diminishing-returns mechanism formalized by Theorem 3.4. □

B Lazy-Greedy Algorithm

Algorithm 1 below gives the lazy-greedy procedure for the exact retained-mass objective. 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 $p_1, \dots, p_n$ over V; budget r; mandatory support B with positive mass
Ensure: Selected support A with $|A| = r$
$M_i \leftarrow \sum_{v \in B} p_i(v)$ for all $i = 1, \dots, n$
Compute $\Delta(v) \leftarrow \frac{1}{n} \sum_{i=1}^{n} \log\big(1 + p_i(v)/M_i\big)$ 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 $\mathcal{H}$
    $\Delta \leftarrow \frac{1}{n} \sum_{i=1}^{n} \log\big(1 + p_i(v)/M_i\big)$ {recompute exact gain}
    if $\mathcal{H}$ is empty or $\Delta \geq$ 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

This algorithm states the exact retained-mass objective. For the smoothed empirical variant, initialize M_i with $\beta + \sum_{v \in B} p_i(v)$ and use the sparse teacher representation described in Appendix C.2. Lazy greedy is exact for whichever objective and initialization are supplied.

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: $p_i \sim \mathrm{Dirichlet}(\alpha \cdot \pi^{(d)})$, 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 $d \in \{0, \dots, D{-}1\}$ receives weight $w_d \propto 1/(d{+}1)$ (harmonic weights). The number of calibration contexts from domain d is $\lfloor w_d \cdot 5{,}000 \rceil$, and simi-

larly 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 Empirical metric and support conventions

The clean quality runs use a sparse, smoothed retained-mass score

$$\widehat{J}_\beta(S) = -\sum_i w_i \log\left(\beta + \sum_{v \in S \cap T_i} p_i(v)\right),$$

where T_i contains the top $8{,}192$ teacher tokens, $\beta = 0.01$, and the evaluation weights sum to one and give each language equal total weight. Smaller values are better. The estimator replaces full teacher probabilities with their top-L entries and applies a floor β for numerical stability, producing a practical surrogate rather than exact reverse KL. This approximation preserves the intended role of the objective as a log-coverage criterion in our implementation; the reported downstream results provide the relevant empirical validation. We compare all methods under the same estimator and do not assume that smoothing preserves their ordering under exact KL. Full-support values marked $0^\dagger$ are analytical KL references, not outputs of the smoothed estimator; at complete mass the latter would be $-\log(1+\beta)$.

The clean selector initializes coverage with the measured mandatory-token mass plus β. Mandatory tokens are model-specific: the implementation includes special and added tokens, plus singleton encodings of byte-fallback token strings. A common mandatory set does not imply constant teacher probability mass across contexts. In these quality runs, K counts the complete retained set, including mandatory tokens. The sample-based acceptance study, where present, uses a separate convention documented with its results. FR-Spec budgets count the drafter-head rows.

The original figures retain their reported axis labels and values. Their generating scripts and complete run records are not included in the available artifact bundle, so their numerical agreement with the clean runs is not established. We distinguish these earlier figures from the clean result tables and do not pool their values.

C.3 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. The multilingual analysis uses Qwen3.5-0.8B-Base and ten FLORES-200 languages spanning Latin, Cyrillic, Arabic, Devanagari, and CJK scripts. Calibration uses the `dev` split and evaluation uses `devtest`, yielding roughly 1,000 contexts per language and split; each context stores the top 8,192 teacher probabilities.

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). The original study reports approximately $1{,}000$ contexts per language and split. Its exact extraction records are not archived with the available figures; the independently documented clean protocol uses one distribution at the end of each sentence's first 60% of tokens. 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.4 Multilingual text continuation

Generation protocol. The original continuation study reports prompts from the first 60% of sentence tokens and continuations of up to 80 tokens using greedy (argmax) decoding. The complete original sampling records are not available; the clean quality runs cap generation at 200 sentences per language. 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.

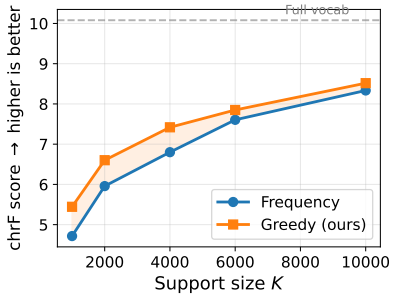
C.5 Robustness across deployment types

Deployments. The robustness study (Table 2) 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. These clean runs extract one next-token distribution at the end of each eligible sentence's first 60% of tokens, storing the top $8{,}192$ probabilities. FLORES-10 contains $9{,}856$ calibration and $10{,}039$ evaluation contexts after filtering; chrF generation uses up to 200 sentences per language. The earlier figures have separate, incompletely archived run records (Appendix C.2); we do not infer their precise extraction settings from numerical differences or compare absolute values across these runs.

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 Experimental Results

The controlled setup and multilingual distributional analysis now appear in the main paper. We retain two complementary results here: downstream continuation quality and the diagnostic connecting the empirical gap to disagreement in token rankings.



K	Method	en	fr	ar	ja	ru	zh	Avg
Full	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**

Figure 4: Multilingual continuation under English-dominant calibration. Left: macro chrF versus support size. Right: per-language chrF at selected budgets. Improvements are largest for scripts that frequency underrepresents.

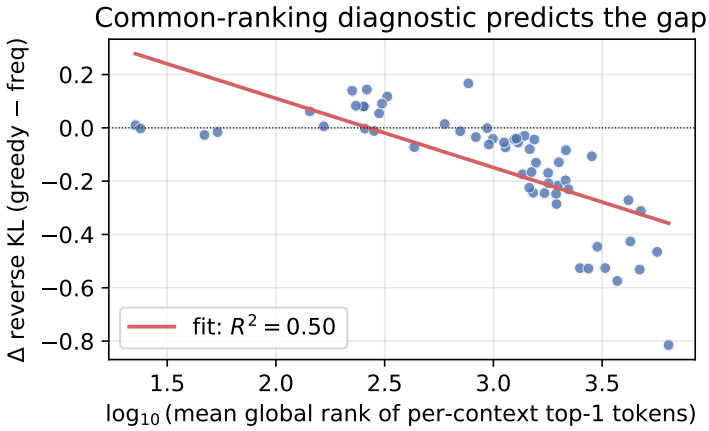


Figure 5: Common-ranking diagnostic across evaluated conditions. As the globally ranked positions of per-context top tokens become worse, the greedy-minus-frequency reverse-KL difference becomes more negative, meaning a larger greedy advantage. A linear fit explains half of the observed variation.

E Sample-Based 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*. For a fixed context, when the drafter equals the target distribution restricted to the selected support, its acceptance probability under Leviathan–Chen sampling (Leviathan et al., 2023; Chen et al., 2023) equals the retained target mass, and its reverse KL equals the negative logarithm of that probability. Our objective therefore maximizes geometric-mean retained mass over the calibration distribution; whether this yields higher aggregate acceptance during decoding is an empirical question, which we test here.

We use the same four calibration mixtures as Table 2, with balanced evaluation across their languages. Qwen3.5-0.8B-Base serves as target and drafter. This runner selects r=1,000 tokens before adding mandatory tokens: the logged greedy and frequency supports contain 1,033 tokens, whereas the magnitude supports contain 1,032. They are not the exact K=1,000 total-token supports of the clean quality tables. At each cycle, the drafter proposes γ=4 tokens, with target verification using $\min(1, p/q)$ and residual sampling from $(p - q)_+$ on rejection. Acceptance is the mean per-prompt fraction of verified candidates accepted. Evaluation uses the corresponding corpus for each mixture, n=50–200 prompts per cell, with stratified paired-bootstrap 95% CIs.

Table 5 reports balanced FLORES-10 evaluation: greedy gains +1.4pp overall (p=0.004), with gains on non-Latin scripts except Hindi and no statistically resolved English difference. Under the

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	$+3.8$	$+1.3$	$\mathbf{+1.4}$

Table 5: Per-language sample-based acceptance with 1,033 retained tokens for both greedy and frequency (1,000 selected before adding mandatory tokens), Qwen3.5-0.8B-Base, FLORES-200, 20 prompts per language. The balanced gain is $+1.4$pp (p=0.004). Hindi reverses by -11.6pp; these data do not isolate a tokenizer mechanism. Calibration and evaluation weights are distinct. The aggregate matrix and draft-length sensitivity are in Appendix E.1.

99%-English/1%-Arabic *calibration* mixture, evaluation still uses 40 English and 40 Arabic prompts. The resulting balanced gain is $+3.3$pp (95% CI $[+1.6, +4.9]$); the Arabic gain is $+7.2$pp and the English difference is -0.7pp. Reweighting these same per-prompt estimates to $99/1$ gives -0.6pp (CI $[-2.9, +1.6]$), so the balanced result does not establish a gain on a 99%-English evaluation stream. The Latin-5 and script-5 aggregate differences are within bootstrap uncertainty. Gemma's different Hindi result (Appendix G.1) shows model/tokenizer dependence but does not isolate segmentation as the cause. The connection between the quality score and aggregate acceptance is empirical.

E.1 Full-matrix aggregate and γ sweeps

Table 6 summarizes balanced evaluation for each calibration mixture, including the magnitude baseline. The high-skew label describes calibration only; its evaluation is equally divided between English and Arabic.

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 $[-2.2, +0.9]$
99/1 calibration; 50/50 eval	0.05	0.37	**0.40**	$+3.3$ $[+1.6, +4.9]$

Table 6: Balanced aggregate acceptance by calibration mixture. Greedy and frequency retain 1,033 tokens; magnitude retains 1,032. CIs use within-language paired resampling of per-prompt acceptance differences. The high-skew row uses $99/1$ calibration but $40/40$ English/Arabic evaluation. The correspondence with the quality-score differences is an empirical observation, not a consequence of a bound on aggregate acceptance.

Draft length sensitivity. We repeat the $99/1$-calibrated bilingual experiment at $\gamma \in \{2, 4, 8\}$, keeping evaluation balanced at 40 prompts per language. The balanced greedy–frequency gap increases from $+3.4$pp to $+4.3$pp in these runs. This supports robustness of the observed balanced gain across the tested draft lengths, without identifying a causal explanation for the trend.

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: Balanced English/Arabic acceptance after $99/1$ calibration, at three draft lengths. Each method retains 1,033 tokens. Δ is the mean per-prompt difference with a stratified paired-bootstrap 95% interval, 40 prompts per language. These are not $99/1$-weighted evaluation gains.

Eval text	Drafter vocabulary	$\bar{\ell}$	tok/s	vs. full
English	Full (no pruning, V=128k)	2.550	144.1	—
	FR-Spec frequency (K=16k)	2.349	211.3	+46.6%
Multilingual	Full (no pruning, V=128k)	1.657	94.0	—
	FR-Spec index (English SlimPajama, K=16k)	1.418	127.9	+36.0%
	Deployment-matched greedy (K=16k)	1.406	126.9	+35.0%

Table 8: **(a) Pruning the drafter vocabulary yields large end-to-end gains at K=16k, where measured throughputs are similar.** Llama-3.2-1B-Instruct target with its shipped 1-layer EAGLE-2 drafter; pruning restricts only the *drafter*'s output vocabulary (the target keeps full V). $\bar{\ell}$ = mean accepted length; tok/s = measured wall-clock throughput; "vs. full" compares each pruned drafter to the no-pruning baseline on the same eval text (n=12 English and n=18 multilingual prompts, two per language across nine languages; 256 generated tokens). Pruning the drafter to K=16k trades a small amount of accepted length for a much cheaper draft step, increasing throughput by +47% on English and +36% on multilingual text over full-vocabulary drafting. At this large budget the English-derived FR-Spec index and a deployment-matched greedy vocabulary are within 1% of each other, a descriptive timing comparison rather than evidence that the supports are identical; the effect of the *selection rule* appears at smaller budgets and is quantified with paired significance in panel (b) and against the full-vocabulary ceiling in panel (c).

F End-to-End Speculative Decoding: Full FR-Spec/EAGLE-2 Results

This section reports the full measurements behind Table 4: an end-to-end speculative-decoding study in the FR-Spec/EAGLE-2 pipeline (Zhao et al., 2025; Li et al., 2024b), pruning only the drafter's output vocabulary while the target keeps the full vocabulary. For the FR-Spec frequency baseline, we replace the teacher-predictive marginal with corpus counts, an inexpensive proxy that estimates the same deployment marginal when the calibration labels are representative samples from the teacher distribution. The pretrained EAGLE-2 drafter keeps full input embeddings and has a separate output matrix; our FR-Spec implementation physically compacts only this draft output head, so its removed rows yield real parameter-memory savings. Two questions must be kept separate: the *value of pruning at all* (does a smaller drafter head decode faster?) and the *value of greedy over frequency* at a fixed budget. Table 8 shows that pruning the drafter yields large wall-clock gains at a large budget where measured throughputs are similar; Table 9 shows that at equal budget K greedy beats frequency with paired significance; and Table 10 adds the no-pruning baseline, exposing the LM-head bottleneck—the full-vocabulary drafter maximizes accepted length yet minimizes throughput—and showing that greedy recovers more of the acceptance ceiling than frequency across two model families. All measurements in this section use the targets' shipped EAGLE-2 drafters on a single NVIDIA A10G GPU in fp16, with wall-clock timing measured around each generation call. Panels (b) and (c) use held-out Tatoeba prompts; the K=16k study in panel (a) uses a fixed set of 12 English and 18 multilingual authored prompts (two per language across nine languages).

G Additional Cross-Family Results

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

To verify that the qualitative pattern of Figure 3 is not specific to the Qwen tokenizer family, we repeat the ten-language distributional comparison on Gemma-3-1B-PT (Gemma Team, 2025) (V=262,144), using the ten FLORES-200 languages of Appendix C.3 ("FLORES-10" hereafter) and the graduated English-dominant calibration weights of Table 2. Table 11 reports global $\widehat{J}_\beta$ at three support budgets and Table 12 reports the per-language breakdown at K=10,000.

G.2 Cross-family quality on Llama-3.2-1B

Table 13 repeats the quality comparison on Llama-3.2-1B ($|V|$=128,256), adding a third model family to Table 1. Evaluation uses Tatoeba prompts rather than FLORES; both are multilingual

Setting	$\bar{\ell}_{\text{freq}}$	$\bar{\ell}_{\text{greedy}}$	tok/s (freq)	tok/s (greedy)	greedy win/tie/lose
Per language, K=1,000*, en* 0.99 / ⟨*lang*⟩ 0.01 (n=40 *prompts; all* $p<0.01$ *Wilcoxon)*					
Arabic (ar)	1.139	1.233	111.1	117.9	33 / 6 / 1
Russian (ru)	1.372	1.402	130.9	133.4	19 / 19 / 2
Chinese (zh)	1.159	1.185	114.7	117.0	20 / 20 / 0
Japanese (ja)[†]	1.277	1.295	124.4	125.9	14 / 21 / 5
Hindi (hi)	1.543	1.558	148.6	149.7	20 / 13 / 7
Vocabulary budget K *(Arabic eval, en* 0.99*,* n=80 *prompts)*					
K=500	1.084	1.136	107.1	111.9	57 / 22 / 1
K=1000	1.145	1.253	111.1	117.9	63 / 15 / 2
K=2000	1.347	1.400	127.3	130.5	47 / 19 / 14
Cross-model: Qwen2-7B (V=152*k), Arabic,* K=1,000 (n=40 *prompts)*					
Qwen2-7B	1.285	1.338	28.8	29.7	34 / 2 / 4

Table 9: **(b) Greedy vs. frequency drafter vocabulary at equal budget.** Same pipeline as (a); here both columns use a pruned drafter of *identical* size K, differing only in *which* tokens are kept (frequency vs. our greedy). $\bar{\ell}$ = mean accepted length from the accepted-length runs (n=80 for the K-sweep, n=40 elsewhere); tok/s = wall-clock throughput from separate paired timing runs on the same cells. Because the two column families come from different runs, relative gains computed across them can differ (e.g. K=1,000: $+9.4\%$ accepted length vs. $+6.1\%$ throughput); within a single timing run, throughput gains track accepted-length gains to within 0.4 pp. "greedy win/tie/lose" counts, over the n paired prompts, on how many greedy achieved a higher / equal / lower accepted length than frequency (each prompt is run under both rules); these paired tests concern accepted length, and the throughput columns are descriptive. Greedy wins on every language and both other axes; all five languages clear Bonferroni correction ($\alpha/5$=0.01) under a one-sided paired Wilcoxon test, and the direction is order-invariant (verified by reversing run order and against a fresh unseen prompt set; a frequency-vs-frequency negative control gives 0.04%). [†]Japanese is borderline: it clears Wilcoxon (p=0.005) but fails the stricter paired t-test (p=0.014) and sign test. The per-language numbers isolate the mechanism on minority-language evaluation; these language-specific gains are not whole-stream throughput estimates for a 99%-English deployment.

parallel corpora, and since every method is evaluated on identical prompts, the relative comparison is unaffected. The headline effect is the high-skew cell, where greedy keeps Arabic usable while frequency collapses it (a $24\times$ per-language chrF gap at K=1,000).

H Calibration-Skew Sweep on FLORES-10

Figure 3 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 14 reports global $\widehat{J}_\beta$ for each method.

		Accepted length $\bar{\ell}$			Throughput (tok/s)			
Model	Lang	Full	Freq.	Greedy	Full	Freq.	Greedy	Δtok/s
Llama-3.2-1B	ar	1.363	1.129	1.199	77.4	111.3	**117.8**	+5.9%
	ru	1.607	1.337	1.362	90.7	130.9	**133.7**	+2.1%
	zh	1.531	1.165	1.190	87.0	114.5	**117.0**	+2.2%
	ja	1.399	1.266	1.283	79.5	124.2	**125.9**	+1.4%
	hi	1.580	1.520	1.532	89.6	148.3	**149.8**	+1.0%
Qwen2-7B	ar	1.491	1.265	1.306	25.4	28.8	**29.7**	+3.1%
	ru	1.592	1.381	1.381	27.1	31.3	31.4	tie
	zh	1.694	1.243	1.263	**28.9**	28.3	28.7	+1.6%
	ja	1.512	1.337	1.336	25.8	30.4	30.4	tie
	hi	2.270	2.041	2.091	38.4	45.8	**46.9**	+2.5%

Table 10: **(c) Full-vocabulary baseline: pruning trades a little acceptance for a large throughput gain, and greedy recovers more of the ceiling than frequency.** Same FR-Spec/EAGLE-2 pipeline as (a)–(b); K=1,000, en 0.99 / $\langle$lang$\rangle$ 0.01 calibration, n=40 held-out prompts per cell. $\bar{\ell}$ = mean accepted length; tok/s = measured throughput; Δtok/s is greedy relative to frequency (computed from unrounded values). The full-vocabulary (no-pruning) drafter attains the *highest* accepted length but the *lowest* throughput on almost every cell: its unpruned 128k/152k-wide LM head dominates per-step cost, so pruning to K=1,000 sacrifices a little acceptance for a large per-step speedup (e.g. Llama-3.2-1B Arabic $77.4 \rightarrow 117.8$ tok/s). This isolates the *value of pruning at all*, complementary to the *value of greedy over frequency* quantified with paired significance in panel (b). At equal budget, greedy recovers more of the full-vocabulary acceptance ceiling than frequency (Llama Arabic 88% vs. 83% of full $\bar{\ell}$) and is the faster rule in every cell except two Qwen ties (Russian, Japanese); Chinese is the single cell where pruning does not beat the full-vocabulary drafter on throughput (28.7 vs. 28.9 tok/s), which we report rather than omit. Values are means from an independent consolidated run used for the full-vocabulary column and the Qwen2-7B breadth check; the paired significance in panel (b) comes from a separate run, so its $\bar{\ell}$ differ by 1–3% (throughput agrees to within 0.3 tok/s).

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 11: Global $\widehat{J}_\beta$ 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 12: Per-language $\widehat{J}_\beta$ 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 Figure 3 (right). Frequency is slightly better on three Latin-script languages; greedy is strictly better on every non-Latin script.

Method	K	$\widehat{J}_\beta$	chrF
Latin-5 (low heterogeneity)			
Full vocab	—	$0^\dagger$	19.84
Greedy	1000	1.317	13.26
Frequency	1000	1.354	12.83
Magnitude	1000	4.461	5.19
Greedy	4000	0.653	17.05
Frequency	4000	0.647	17.15
Magnitude	4000	3.797	6.55
Multi-script-5 (disjoint scripts)			
Full vocab	—	$0^\dagger$	15.10
Greedy	1000	1.416	8.72
Frequency	1000	1.463	8.82
Magnitude	1000	4.502	0.85
Greedy	4000	0.582	12.44
Frequency	4000	0.580	12.12
Magnitude	4000	4.230	1.04
High skew (en+ar 0.99/0.01)			
Full vocab	—	$0^\dagger$	14.42
Greedy	**1000**	**1.571**	**9.18**
Frequency	1000	2.386	6.11
Magnitude	1000	4.445	1.93
Greedy	4000	0.625	12.64
Frequency	4000	0.791	12.09
Magnitude	4000	4.096	2.18

Table 13: **Cross-family quality on Llama-3.2-1B** ($|V|$=128,256): $\widehat{J}_\beta$ (lower is better) and macro-averaged chrF (higher is better) for three calibration mixtures, at $K \in \{1000, 4000\}$. Evaluation uses Tatoeba prompts, shared by greedy and frequency within each cell. The Latin-5 differences are small relative to those in the high-skew cell; this does not establish the common-ranking condition. On Multi-script-5, the score favors greedy at K=1000 and frequency at K=4000; chrF favors the opposite method at each budget. The largest greedy advantage occurs after high-skew English/Arabic calibration: at K=1000, Arabic chrF is 6.55 for greedy versus 0.27 for frequency. These chrF values alone do not establish qualitative usability. The score is the common smoothed surrogate in Appendix C.2, not exact KL; $0^\dagger$ denotes an analytical full-support KL reference.

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 14: Global $\widehat{J}_\beta$ 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 3.6. Magnitude is independent of calibration weights, so its value is constant across rows.