LET THE MODEL CHOOSE: SELECTING SAMPLERS FOR DIVERSE ANSWERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Best-of-N selection, pass@k evaluation, and offering users alternatives all draw several answers from one language model. Independent sampling spends part of this budget on repeats. Stratified arithmetic sampling coordinates the answers instead. Each answer decodes one random number drawn from its own slice of the unit interval, so the set spreads over the model's options while every answer keeps the model's distribution. Kept for the whole answer, this number is rescaled at every token. Unfortunately, that rescaling operation leads to accumulated precision errors, and in float64 it runs out of precision after a median of 44–79 tokens of a real response. The rest of the text is then chosen by rounding error. This, and similar issues, have prevented more widespread adoption of arithmetic sampling and similar methods. To overcome this flaw, we introduce *early release*: an answer switches back to ordinary sampling as soon as its prefix interval lies inside its slice. We prove that early release produces exactly the joint distribution of answer sets of infinite-precision stratified arithmetic sampling, and that with eight answers it needs more than float64's 53 bits with probability below 2^{-50}. In practice it releases after a median of 6–12 tokens and runs 14% faster than always-on arithmetic sampling. It composes with top-p, top-k, min-p, and p-less. Across models from 360M to 70B parameters, early release finds 11.4% more distinct correct answers on CoverageQA over a grid of four truncation rules and four temperatures, raises pass@8 on all 11,313 TriviaQA validation questions from 34.72% to 35.93%, and yields 3.6% more quality-approved meanings on NoveltyBench, with no detectable change in the quality of individual answers.

1 INTRODUCTION

Many uses of a language model draw several answers to one prompt. Best-of-N selection and repeated sampling need a correct candidate for a verifier to pick (Cobbe et al., 2021, Li et al., 2022, Brown et al., 2024, Snell et al., 2025, Wu et al., 2025). Pass@k scores a model by whether any of k attempts succeeds (Kulal et al., 2019, Chen et al., 2021). An assistant asked for alternatives should return distinct ideas (Zhang et al., 2025). In each case the value lies in what the *set* of answers contains. Independent sampling ignores the set. Each answer is drawn afresh, so probable answers repeat and less probable ones are missed, a problem that grows as aligned models concentrate probability mass (Kirk et al., 2024, Padmakumar & He, 2024, West & Potts, 2025, Jiang et al., 2025).

Coordinated sampling links the random choices behind the answers while keeping each answer distributed like an ordinary sample. Arithmetic sampling (Vilnis et al., 2023) makes this simple. Decoding one answer by inverse-CDF sampling can be driven by a single random number in $[0, 1)$, and nearby numbers produce answers with the same beginning. Giving each of K answers a number from its own slice of $[0, 1)$, a construction known as stratified sampling (McKay et al., 1979), spreads the set across the model's options (Figure 1, left). QuasiMoTTo applies this idea and related quasi-random designs to language models (Li et al., 2026).

Always-on arithmetic sampling has a practical flaw. The single number is rescaled after every token, and each token consumes about $-\log_2 p$ of its bits (Witten et al., 1987, Moffat et al., 1998), like a zoom that reads off the leading digits of the number (Figure 2). A float64 number holds 53 bits, about 16 decimal digits. On real responses the number is used up after a median of 44–79 tokens, before 47–85% of short open-ended answers and 93–95% of long ones end (Appendix B.1). From then on,

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.

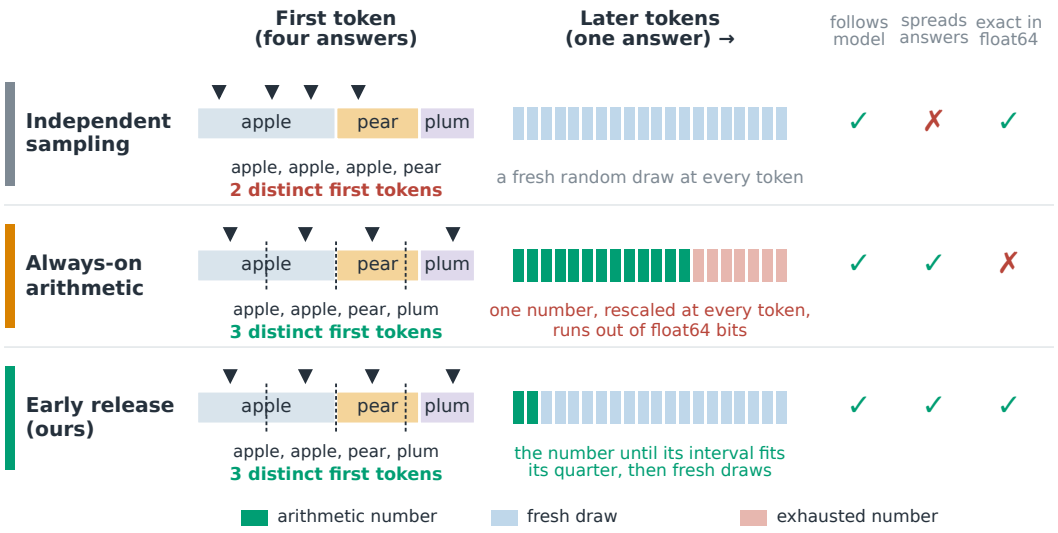

Figure 1: **Three ways to draw four answers from the same model.** In this toy example the first token is apple, pear, or plum with probabilities 50%, 30%, and 20%, laid out as intervals on $[0, 1)$. A random number selects the option whose interval contains it. *Independent sampling* draws four unrelated numbers, which can repeat an option. *Stratified arithmetic sampling* draws one number from each quarter of $[0, 1)$, spreading the answers. Always-on arithmetic sampling keeps rescaling it for every later token, and in float64 it runs out of bits (Figure 2). *Early release* keeps the number only until the answer's prefix interval lies inside its quarter, then draws fresh numbers. It has the same joint distribution of answer sets as exact stratified arithmetic sampling (Theorem 1).

tokens are selected by accumulated rounding error instead of the random number. The sampler no longer implements its stated distribution, and its output depends on numerical details of the hardware (Yuan et al., 2025). A perturbation of 10^{-4} in the logits changes 96% of its responses, and on some sources it collapses entirely. Keeping the number also forces a cumulative distribution and a sort over the vocabulary at every step.

The coordination is needed only briefly. Once the tokens generated so far pin an answer inside its assigned slice, the slice no longer constrains anything, and the rest of the answer can be sampled ordinarily. We call this switch *early release* (Figure 1, third row). Our contributions are:

- **Early release.** Releasing an answer as soon as its prefix interval lies inside its slice reproduces exactly the joint answer-set distribution of infinite-precision stratified arithmetic sampling. It is the earliest release valid for every model, and with eight answers it needs more than 53 bits with probability below 2^{-50}.
- **Float exhaustion.** On real responses, always-on arithmetic sampling exhausts float64 within 44–79 tokens, while early release stops after about 5.1 bits and runs 14% faster.
- **A broad evaluation.** From 360M to 70B parameters and across four truncation samplers, early release adds correct alternatives on CoverageQA, TriviaQA, and NoveltyBench without changing individual quality, and leaves math, code, and reasoning-model accuracy unchanged.

2 Coordinated Sampling with Early Release

Fix a prompt and a decoding configuration: temperature, a truncation rule such as top-p (Holtzman et al., 2020), top-k (Fan et al., 2018), min-p (Nguyen et al., 2025), p-less (Tan et al., 2026), or other truncation rules (Hewitt et al., 2022, Meister et al., 2023), and a length limit. These define a distribution P over complete responses. Every method below draws from the same truncated and renormalized next-token probabilities, so it composes with any of these rules.

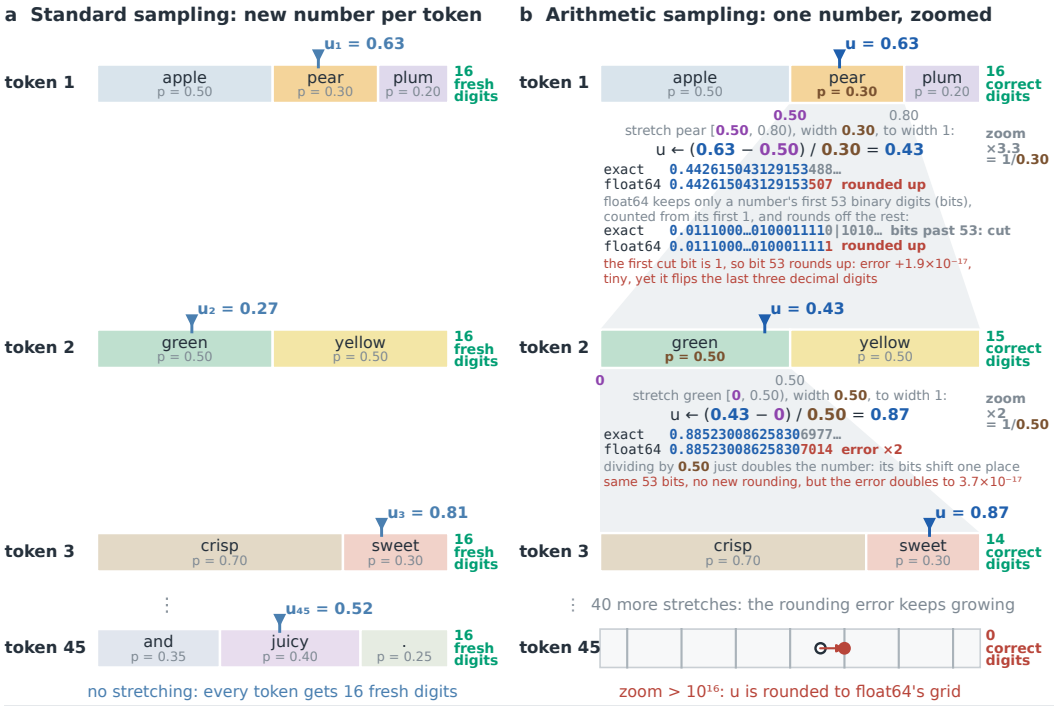

Figure 2: **Why standard sampling never runs out of precision and always-on arithmetic sampling does. (a)** Standard sampling draws a new random number u_t for every token and reads it on the unstretched next-token bar, so every token has all 16 significant digits available. Each bar lists the next-token probabilities p. **(b)** Arithmetic sampling reuses one number u. Colors track each value: u in blue, including its exact and float64 expansions, the chosen interval's left end in purple, and its width, which is the token's probability, in brown. All numbers shown are float64 values shortened to two digits. Token 1 is pear because u lies in pear's interval $[0.50, 0.80)$. To pick token 2 with the same number, that interval is stretched to width 1, $u \leftarrow (u - 0.50)/0.30$, a zoom by $1/0.30 \approx 3.3$ that magnifies every distance, including any error. Float64 keeps only a number's first 53 binary digits (bits), counted from its first 1, which is about 16 decimal digits. The exact result needs more, so float64 cuts it after bit 53 and, because the first cut bit is 1, rounds up. The error is only 1.9×10^{-17} yet changes the last three printed decimal digits. Stretch 2 divides by 0.50, which just doubles the number and shifts its bits one place: nothing new is rounded, but the earlier error doubles. Each zoom by $1/p$ magnifies the accumulated error by $1/p$ and costs about $\log_{10}(1/p)$ correct digits. After about 45 tokens of a real response the zooms multiply past 10^{16}, no correct digit is left, and rounding decides every later token. Early release keeps u only until the answer's slice is settled, about 1.5 digits in our measurements (Appendix B.1), then gives every later token a fresh number, as in (a).

2.1 One random number can generate a whole answer

Inverse-CDF sampling. Order the vocabulary and lay the next-token probabilities side by side on $[0, 1)$. A number u drawn uniformly from $[0, 1)$ selects the token whose interval contains u. Ordinary decoding repeats this with a fresh u at every step.

Arithmetic sampling. The same u can also choose the later tokens (Vilnis et al., 2023), as in arithmetic decoding (Rissanen & Langdon, 1979, Witten et al., 1987). After selecting a token with interval $[a, a + p)$, subdivide it by the next-token probabilities and select again with the same u. The interval left after a prefix h, written $I(h) = [L, L + W)$, contains exactly the numbers that generate h, and its width W equals the probability of h. In floating point, the number stays in $[0, 1)$ by rescaling, $u \leftarrow (u - a)/p$, after each token. A uniform u yields a response with distribution P.

Stratified answer sets. To draw K answers, split $[0, 1)$ into K equal slices $S_1, \ldots, S_K$, draw u_j uniformly and independently inside S_j, decode one answer from each, and shuffle the answers. Each answer still has distribution P, since a random slice with a uniform point inside is a uniform point. The answers are no longer independent: they start in different regions of the response space. For any fixed notion of success, such as "the answer is correct", let q_j be the success probability of the answer decoded from slice j. The single-answer success rate is their mean $\bar{q}$, and independence across slices gives

$$\Pr\{\text{some answer succeeds}\} = 1 - \prod_{j=1}^{K}(1 - q_j) \;\geq\; 1 - (1 - \bar{q})^K, \tag{1}$$

the value for K independent answers, by the arithmetic–geometric mean inequality. Expected pass@K therefore never decreases, and it increases strictly when success depends on the slice. The same argument, applied per class, shows that the expected numbers of distinct correct answers and of fixed semantic classes covered never decrease (Appendix A).

2.2 Always-on arithmetic sampling exhausts float64

Rescaling spends precision. A token of probability p stretches the number by $1/p$ and uses about $\log_2(1/p)$ of its bits; float64 keeps 53 (Figure 2). On real responses the number is used up after a median of 44–79 tokens, and 47–85% of curated NoveltyBench responses outlast it. Accumulated rounding error then selects the tokens. The output also becomes hypersensitive: a logit perturbation of 10^{-4} changes 96% of always-on responses versus 3% for independent sampling (Appendix B.1).

2.3 Early release

Follow one answer with slice S_j and current prefix interval $I(h)$. While $I(h)$ crosses a boundary of S_j, the slice still restricts which continuations are possible. Once $I(h) \subseteq S_j$, every number that produces h already lies in the slice. We release the answer at that point:

$$I(h) \subseteq S_j \quad \Longrightarrow \quad \text{continue with fresh random numbers.} \tag{2}$$

Theorem 1 (Early release is exact)**.** *Given its slice and a prefix h with $I(h) \subseteq S_j$, the rescaled number $(u - L)/W$ is uniform on $[0, 1)$ and independent of h. Hence replacing it by fresh uniform draws preserves the conditional distribution of every continuation. Because slices draw their numbers independently, early release produces exactly the joint distribution of the answer set under stratified arithmetic sampling in exact arithmetic. Moreover, containment is necessary: before $I(h) \subseteq S_j$, some next-token distribution distinguishes the retained number from a fresh one.*

The proof is in Appendix A.3. Any fixed procedure on the answer set, such as a verifier, a semantic grouping, or a diversity score, therefore has the same output distribution under both, and Equation 1 still holds.

Release is fast, so float64 suffices. A prefix that still needs its number must straddle one of the $K - 1$ slice boundaries. On each decoding path, take the first prefix whose interval is narrower than $2^{-b}/K$. These intervals are disjoint, so at most $K - 1$ of them straddle a boundary, with total probability at most $(K - 1)2^{-b}/K$ (Proposition 3). This gives, in exact arithmetic and for any model,

$$\Pr\{\text{an answer is still coordinated after } \log_2 K + b \text{ bits of surprisal}\} \;\leq\; (1 - 1/K)\, 2^{-b}. \tag{3}$$

With $K = 8$, an answer consumes more than 53 bits before release with probability below 2^{-50}. The float64 number therefore still carries unused bits when it is discarded. On real responses release happens after a median of 6–12 tokens and about 5 bits, close to $\log_2 8 = 3$ bits plus a small overhead, and never after more than 17.3 bits in our measurements (Appendix B.1). An ablation (Figure 6, Appendix B.3) confirms that the release point is the right one: releasing after a fixed single token already recovers most of the gain, holding the number longer adds nothing, and storing it in as few as 8 bits leaves early release unchanged. Even for always-on sampling, losing bits leaves pass@1 intact and costs exactness and reproducibility.

Cost. After release an answer uses the engine's ordinary sampler, so the extra cost of coordination is limited to a few tokens. Across 25 complete benchmark runs, early release took 12% longer than independent sampling on average, and 1–4% on long open-ended responses, while always-on stratified sampling took 30% longer (Appendix B.2).

3 EXPERIMENTAL SETUP

Methods. Every comparison fixes the model, prompt, temperature, truncation rule, length limit, and number of answers K, and changes only how the K answers draw their randomness. *Independent* sampling uses the engine's sampler. *Always-on stratified* is full stratified arithmetic sampling with a float64 number. *Lattice* is Arithmetic Sampling with one shared random offset and K equally spaced numbers (Vilnis et al., 2023), a randomly shifted lattice (Cranley & Patterson, 1976). *Early release* is our method. All methods run in vLLM (Kwon et al., 2023) through one logits processor, with the same probability ordering, and use $K = 8$ unless stated otherwise. Each truncation rule has its own matched independent baseline: top-p 0.9, top-k 50, min-p 0.05, and p-less.

Models. We use Qwen2.5 0.5B, 1.5B, 7B, Math 1.5B, and Coder 1.5B and 7B Instruct (Qwen et al., 2025, Yang et al., 2024, Hui et al., 2024), Llama 3.2 1B and Llama 3.3 70B Instruct with FP8 weights and activations (Grattafiori et al., 2024, Meta, 2024, Meta AI, 2024, Red Hat AI, 2024), and SmolLM2 360M and 1.7B Instruct (Ben Allal et al., 2025). The reasoning models are DeepSeek-R1-Distill-Qwen 1.5B and 7B (Guo et al., 2025), and Qwen3 1.7B in thinking mode (Yang et al., 2025).

Tasks and metrics. All benchmarks are used in full. **CoverageQA** asks 145 questions with many valid answers, such as naming a country in a region, and lists the accepted answers (Wong et al., 2025). We count distinct accepted answers per set. **TriviaQA** has one correct answer per question (Joshi et al., 2017). We use all 11,313 questions of the unfiltered validation split, closed-book, and report pass@8, the fraction of questions answered correctly at least once. **NoveltyBench**'s curated split has 100 open-ended requests (Zhang et al., 2025). Two independent LLM judges rate each response, and responses that both rate at least 3 of 4 are grouped by meaning with NoveltyBench's classifier. We count these *quality-approved meanings* per set. For longer outputs we use MATH-500 (Lightman et al., 2024), all 1,319 GSM8K test problems (Cobbe et al., 2021), all 541 IFEval prompts (Zhou et al., 2023), the HumanEval+ and MBPP+ code benchmarks with the extended EvalPlus tests (Chen et al., 2021, Austin et al., 2021, Liu et al., 2023), and for reasoning models the 60 AIME 2024 and 2025 problems. Appendix C gives prompts, graders, and judges.

Statistics. Differences are paired by prompt. Intervals come from a paired bootstrap over prompts, keeping each prompt's methods together. For figures with several model and filter settings, we also report Bonferroni-adjusted intervals at the familywise 95% level. Averages weight models equally.

4 RESULTS

Figure 3 gives examples, and Figure 4 summarizes the gains on the three set-level benchmarks for every model.

4.1 MORE DISTINCT CORRECT ANSWERS IN THE SAME BUDGET

We compare early release with independent sampling on all 145 CoverageQA questions for seven models from 360M to 70B parameters (Figure 5), four truncation rules, and four temperatures, each with three fresh seeds (1,559,040 responses). Averaged over the grid, early release raises the number of distinct accepted answers per eight-answer set from 1.371 to 1.528, a gain of 11.4% (paired difference +0.157, 95% interval [+0.132, +0.183]), and the gain is significant for every model (Figure 4a). It is positive in 103 of the 112 settings and significant after adjustment in 65. No setting shows a significant decrease, and the 5 exact ties occur where the model is nearly deterministic or almost never gives an accepted answer. Individual answers are accepted as often as before (difference -0.12 points, 95% interval [-0.36, +0.12], Appendix D.1).

Raising the temperature is the usual way to get more alternatives, and it costs accuracy. For Qwen2.5 0.5B, early release at temperature 1 matches the coverage of independent sampling at temperature 1.5 (1.63 versus 1.64 distinct answers) while keeping 46% of answers accepted instead of 37%. Over the grid, always-on stratified sampling reaches 1.529 distinct answers, matching early release as Theorem 1 predicts, at higher cost and without exactness in float64. The lattice of Arithmetic Sampling reaches 1.564, slightly more on this average, but its shared offset makes the numbers

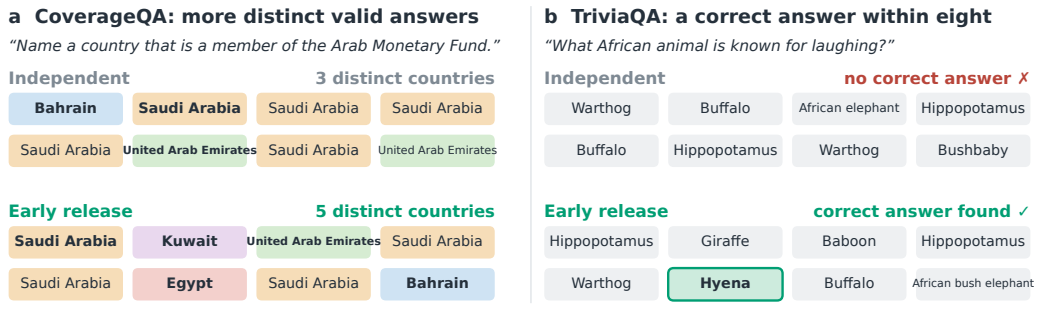

Figure 3: **Eight real answers with independent sampling and with early release.** Qwen2.5 7B, min-p 0.05, temperature 1, with the same prompt and model distribution for both methods. **(a)** On a CoverageQA question, independent sampling repeats Saudi Arabia five times, while early release spreads the same budget over five accepted countries. Chips of one color are the same country; first occurrences are bold. It is one of the 24 of 145 questions where early release finds more accepted answers in this setting (11 where it finds fewer). **(b)** On a TriviaQA question, only early release reaches the correct answer. The question is drawn at random among the 326 questions that early release answers and independent sampling misses in this setting, against 137 the other way round.

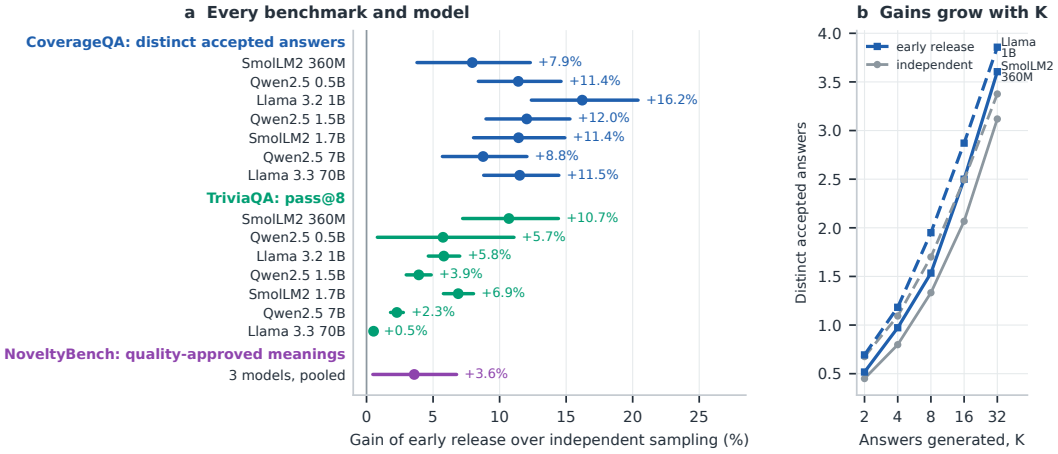

Figure 4: **Early release helps on every set-level benchmark and across model scales. (a)** Relative gain over independent sampling with the same model, truncation rule, and temperature, with paired prompt-bootstrap intervals. CoverageQA counts distinct accepted answers over the full temperature grid (95%). TriviaQA reports pass@8 on all 11,313 validation questions, averaged over truncation rules (95%). NoveltyBench counts quality-approved meanings, pooled over three models as in its preregistered comparison (97.5%). **(b)** Distinct accepted CoverageQA answers for newly generated sets of K answers, averaged over four truncation rules (separate study on two small models, Appendix D.1).

dependent, so it lacks the pass@K guarantee of Equation 1 and can lose to independent sampling on other events (Appendix A.5). Like always-on decoding, it also exhausts float64.

For newly generated sets of 2 to 32 answers, the absolute gain grows with the budget (Figure 4b).

4.2 A CORRECT ANSWER WITHIN THE SET

Equation 1 predicts higher pass@K whenever success depends on the region of the response space an answer starts in, as for factual recall. On all 11,313 TriviaQA validation questions, early release raises pass@8 for all seven models from 360M to 70B parameters (Figure 4a). Averaged over models and four truncation rules, pass@8 rises from 34.72% to 35.93% (paired difference +1.21 points, 95% interval [+1.10, +1.32]). The gain is positive in 28 of the 28 settings and significant after adjustment

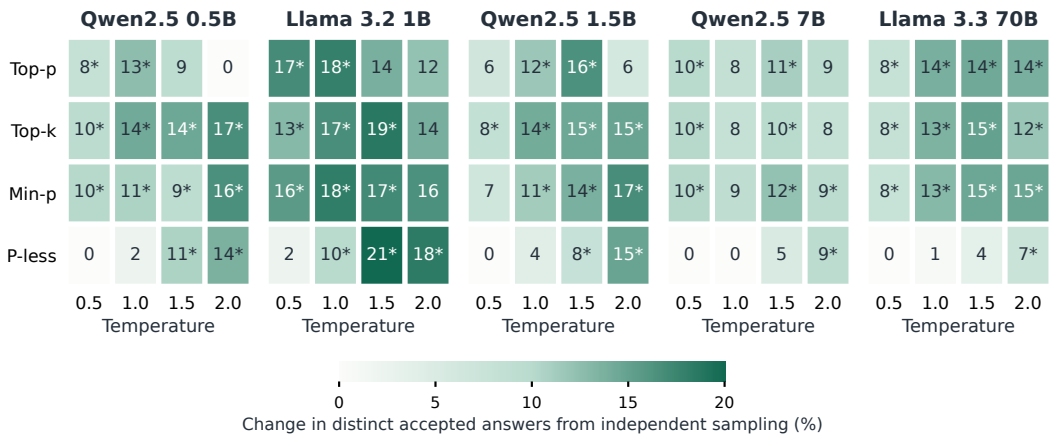

Figure 5: **Early release finds more distinct accepted answers across models, truncation rules, and temperatures.** Each cell gives the change in distinct accepted answers per eight-answer set relative to independent sampling with the same model, rule, and temperature, on all 145 CoverageQA questions and averaged over three fresh seeds. An asterisk marks an interval that excludes zero after Bonferroni adjustment over all 112 cells of the full grid. The two SmolLM2 models, whose answers are rarely accepted at high temperature, are shown with all other models in Figure 9 (Appendix D.1).

in 22. No setting shows a significant decrease. Gains are smallest with p-less, which truncates most aggressively and leaves the fewest alternatives to spread over. Single-answer accuracy is unchanged, since each answer keeps the model's distribution (Appendix D.2).

Longer outputs behave differently (Table 1, Appendix D.4). On MATH-500, GSM8K, IFEval, HumanEval+, and MBPP+, and for reasoning models on MATH-500 and AIME, early release and independent sampling reach statistically indistinguishable pass@1 and pass@K, as Equation 1 allows. These answers open with formulaic text: a reasoning trace releases within its opening sentences, which restate the problem (median 23 tokens), and a HumanEval+ answer only after copying the given signature and docstring, so the slices are resolved on tokens that rarely decide correctness.

4.3 More useful meanings in open-ended requests

On the 100 curated NoveltyBench requests we generate eight responses of up to 1,024 tokens with Qwen2.5 0.5B and 7B and Llama 3.3 70B, all four truncation rules, and temperatures 1.0 and 1.5. Two independent judges rate every response. Among responses that both judges approve, early release yields 1.902 distinct meanings per set versus 1.836 for independent sampling, a 3.6% increase (97.5% interval $[0.5, 6.8]$%), with the largest gain on the 70B model and none on the 0.5B model. The share of individual responses that both judges approve is statistically unchanged, and on all responses six of seven further diversity measures, including the Vendi score (Friedman & Dieng, 2023), also improve after multiple-comparison adjustment (Appendix D.3).

Inside the model. A logit-lens and sparse-autoencoder analysis of Qwen2.5 7B (Appendix E) shows where early release acts. The token that triggers release is one the model leaves open: on NoveltyBench its final next-token distribution has a median entropy of 1.6 bits, against 0.1 bits for the coordinated tokens before it. Early-release answers also differ internally: they share about 4% fewer SAE features with each other at all seven layers tested, on both CoverageQA and NoveltyBench.

5 Related Work

Coordinated sampling for language models. Arithmetic Sampling decodes each answer from one point of an implicit arithmetic code book and spaces K points on a lattice with one shared random offset (Vilnis et al., 2023), a Cranley–Patterson rotation (Cranley & Patterson, 1976). Quasi-random codes help self-consistency and minimum Bayes risk decoding (Amballa et al., 2025), and QuasiMoTTo compares independent strata, lattices, and other quasi-random designs for test-time

Table 1: **Longer outputs: no detectable loss or gain.** Pass@1 and pass@K (%) with independent sampling and early release, averaged over four truncation rules at temperature 1 and over models: Qwen2.5-Math 1.5B and Qwen2.5 7B for math, Qwen2.5 0.5B and 7B for IFEval, and eight models from 360M to 7B parameters for code, scored with the extended EvalPlus tests. Code averages two independent sets of answers per problem, and the reasoning rows two (MATH-500) or three (AIME). Rows marked R use reasoning models at their recommended top-p 0.95 and temperature 0.6: DeepSeek-R1-Distill-Qwen 1.5B and 7B, and Qwen3 1.7B in thinking mode. Differences are paired by problem, with 95% bootstrap intervals in points. All problems are used.

		Pass@1			Pass@K		
Benchmark	K	Ind.	Early	Difference	Ind.	Early	Difference
MATH-500	2	74.9	74.8	[-0.9, +0.6]	81.3	81.2	[-0.9, +0.9]
MATH-500	4	74.9	75.2	[-0.3, +0.9]	85.6	85.9	[-0.5, +1.0]
MATH-500	8	75.0	74.9	[-0.5, +0.4]	89.3	89.0	[-0.9, +0.4]
GSM8K	8	88.2	88.3	[-0.1, +0.3]	95.5	95.5	[-0.4, +0.2]
IFEval	8	48.7	48.5	[-0.6, +0.3]	67.6	67.4	[-1.1, +0.7]
HumanEval+	8	43.1	43.3	[-0.1, +0.5]	62.1	62.1	[-0.7, +0.6]
MBPP+	8	44.9	45.1	[+0.0, +0.4]	65.5	65.9	[-0.1, +0.7]
MATH-500, R	8	87.8	87.9	[-0.2, +0.5]	95.9	96.2	[-0.1, +0.7]
AIME 24–25, R	8	38.4	38.7	[-1.0, +1.4]	62.2	63.5	[-0.9, +3.5]

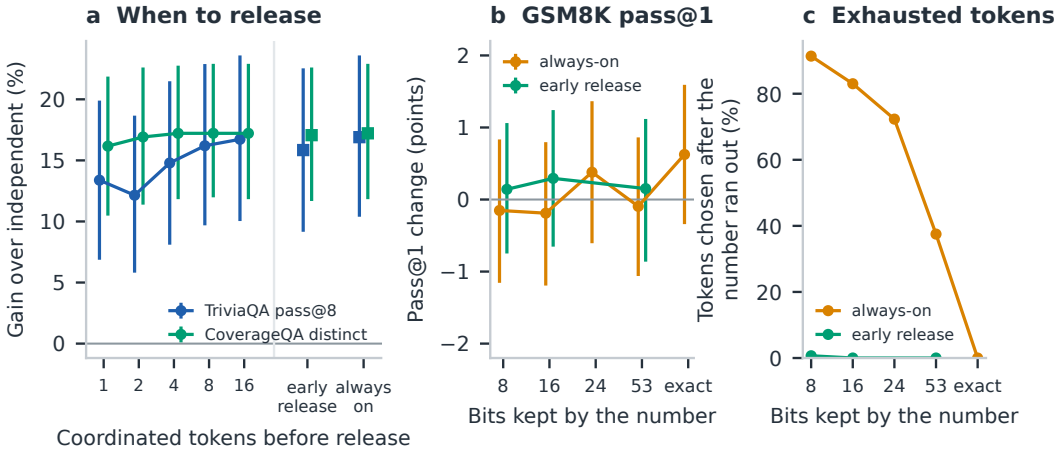

Figure 6: **When to release, and how many bits the number needs.** Qwen2.5 0.5B, min-p 0.05, temperature 1, eight answers. (a) Relative gain over independent sampling in TriviaQA pass@8 (all 11,313 questions) and CoverageQA distinct accepted answers (145 questions, three seeds) when every answer lets go after a fixed number of coordinated tokens, at containment (early release), or never (always-on). (b) Change in GSM8K pass@1 from independent sampling (all 1,319 problems) when the number keeps a given number of significant bits (53 is float64). (c) Share of GSM8K tokens chosen after the number used up its bits. Bars are 95% paired prompt-bootstrap intervals.

scaling (Li et al., 2026). Their arithmetic variants keep the number for the whole answer. We prove that independent strata can drop it at containment without changing the joint law of the answer set, that no earlier rule is valid for every model, and that release time has a bound in bits.

Stratified and quasi-Monte Carlo sampling. Stratified and Latin hypercube sampling place one draw in each cell of a partition, and randomized quasi-Monte Carlo spreads points more evenly while keeping estimates unbiased (McKay et al., 1979, Stein, 1987, Sobol', 1967, Owen, 1997, L'Ecuyer, 2018). Array-RQMC and sequential quasi-Monte Carlo advance many Markov chains with one randomized point set per step (L'Ecuyer et al., 2008, Gerber & Chopin, 2015). We stratify one number per autoregressive chain and stop once its stratum no longer constrains the chain.

Arithmetic coding and finite precision. Arithmetic coding maps a sequence to a nested subinterval of $[0, 1)$ (Rissanen & Langdon, 1979, Witten et al., 1987). Practical coders keep integer intervals and shift out settled leading bits, so a fixed precision serves sequences of any length (Moffat et al., 1998). Language models drive such coders for compression and steganography (Delétang et al., 2024, Valmeekam et al., 2023, Ziegler et al., 2019). A sampler that instead rescales one float64 number per answer, as always-on arithmetic sampling does in our implementation, runs out after a few dozen tokens. Early release plays the role of the settled bits. Once the prefix interval lies in its slice, the remaining digits say nothing about the slice.

Shared randomness and distinct samples. Speculative decoding couples draft and target samples (Leviathan et al., 2023, Chen et al., 2023, Daliri et al., 2025), and distortion-free watermarks drive inverse-CDF sampling with keyed random numbers (Kuditipudi et al., 2024). We couple the answers of one set. Gumbel-top-k, stochastic beam search, and incremental sampling draw sequences without replacement (Maddison et al., 2014, Kool et al., 2019, Meister et al., 2021a, Shi et al., 2020, Grubisic et al., 2024). Diverse and determinantal beam search, CorrSynth, and diversity-steered sampling target set diversity (Vijayakumar et al., 2018, Meister et al., 2021b, Kowshik et al., 2024, Park & Cho, 2025). Their outputs depart from the model distribution, and most are hard to parallelize.

Truncation samplers and output diversity. Top-k, top-p, typical, η, min-p, and p-less sampling reshape one answer's next-token distribution (Fan et al., 2018, Holtzman et al., 2020, Meister et al., 2023, Hewitt et al., 2022, Nguyen et al., 2025, Tan et al., 2026), and other work tunes temperature or learns adaptive decoding (Du et al., 2025, Dhuliawala et al., 2024, Su et al., 2026). Early release composes with any such rule. Alignment reduces diversity, and models converge on similar answers (Kirk et al., 2024, Padmakumar & He, 2024, West & Potts, 2025, Jiang et al., 2025). CoverageQA, NoveltyBench, the Vendi score, and semantic entropy measure it (Wong et al., 2025, Zhang et al., 2025, Friedman & Dieng, 2023, Kuhn et al., 2023, Farquhar et al., 2024). Prompting and training remedies (Zhang et al., 2026, Slocum et al., 2025, Lanchantin et al., 2025) combine with ours.

Repeated sampling and test-time compute. Pass@k asks whether any of k samples succeeds (Kulal et al., 2019, Chen et al., 2021). Code generation, self-consistency, verifier-based best-of-N, and compute-optimal inference all rely on many samples (Li et al., 2022, Wang et al., 2023, Cobbe et al., 2021, Lightman et al., 2024, Brown et al., 2024, Snell et al., 2025, Wu et al., 2025), training can target pass@k directly (Walder & Karkhanis, 2025), and reinforcement learning with verifiable rewards can lower pass@k at large k (Yue et al., 2025). Stratification never lowers expected pass@k for a fixed success criterion and needs no training.

6 Conclusion

Early release keeps coordination only as long as it matters. It reproduces the joint law of exact stratified arithmetic sampling, avoids float64 exhaustion, and costs little more than independent sampling, while adding correct alternatives for models from 360M to 70B parameters. Choosing *where* to coordinate, beyond an answer's start, is a natural next step that early release makes exact.

Limitations. The guarantees concern fixed events and fixed classes. Learned semantic groups and judge ratings depend on the sampled set and on the judges, so we measure them empirically and report individual quality alongside them. Coordination needs access to the sampler's probabilities, which vLLM and similar engines expose through logits processors. Early release coordinates the start of each answer, so tasks whose answers diverge only late gain little from it.

Reproducibility Statement

Section 2 and Appendix A state the sampling rules and prove their guarantees. Appendix B describes the implementation and its tests, and Appendix C gives models, prompts, graders, judges, and statistics. All benchmarks are public and used in full. We will release the logits processor, generation and scoring code, pinned model revisions, all generated responses, judge ratings, and the compact data behind every figure and table.

AI Use Statement

Generative AI tools were used in the preparation of this work, AI-assisted work was reviewed and validated by the authors; evaluation uses of LLMs are described in Appendix C. The authors take responsibility for the final work.

References

Avinash Amballa, Aditya Parashar, Aditya Vikram Singh, Jinlin Lai, and Benjamin Rozonoyer. Quasi-random multi-sample inference for large language models. *arXiv preprint arXiv:2411.06251*, 2025. URL https://arxiv.org/abs/2411.06251.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021. URL https://arxiv.org/abs/2108.07732.

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2025. URL https://arxiv.org/abs/2303.08112.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM2: When Smol goes big — data-centric training of a fully open small language model. In *Conference on Language Modeling*, 2025. URL https://colm.cc/virtual/2025/poster/816.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024. URL https://arxiv.org/abs/2407.21787.

Bart Bussmann, Patrick Leask, and Neel Nanda. BatchTopK sparse autoencoders. *arXiv preprint arXiv:2412.06410*, 2024. URL https://arxiv.org/abs/2412.06410.

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. URL https://arxiv.org/abs/2302.01318.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

R. Cranley and T. N. L. Patterson. Randomization of number theoretic methods for multiple integration. *SIAM Journal on Numerical Analysis*, 13(6):904–914, 1976. doi: 10.1137/0713071. URL https://doi.org/10.1137/0713071.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*, pp. 7827–7845, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/1fa1ab11f4bd5f94b2ec20e794dbfa3b-Paper-Conference.pdf.

Majid Daliri, Christopher Musco, and Ananda Theertha Suresh. Coupling without communication and drafter-invariant speculative decoding. In *2025 IEEE International Symposium on Information Theory (ISIT)*, pp. 1–6. IEEE, 2025. doi: 10.1109/ISIT63088.2025.11195663. URL https://doi.org/10.1109/ISIT63088.2025.11195663.

Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, Marcus Hutter, and Joel Veness. Language modeling is compression. In *International Conference on Learning Representations*, pp. 14165–14181, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/3cbf627fa24fb6cb576e04e689b9428b-Paper-Conference.pdf.

Shehzaad Dhuliawala, Ilia Kulikov, Ping Yu, Asli Celikyilmaz, Jason Weston, Sainbayar Sukhbaatar, and Jack Lanchantin. Adaptive decoding via latent preference optimization. *arXiv preprint arXiv:2411.09661*, 2024. URL https://arxiv.org/abs/2411.09661.

Weihua Du, Yiming Yang, and Sean Welleck. Optimizing temperature for language models with multi-sample inference. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 14648–14668. PMLR, 2025. URL https://proceedings.mlr.press/v267/du25f.html.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 889–898. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1082. URL https://aclanthology.org/P18-1082/.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024. doi: 10.1038/s41586-024-07421-0. URL https://doi.org/10.1038/s41586-024-07421-0.

Dan Friedman and Adji Bousso Dieng. The Vendi score: A diversity evaluation metric for machine learning. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=g97OHbQyk1.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 10835–10866. PMLR, 2023. URL https://proceedings.mlr.press/v202/gao23h.html.

Mathieu Gerber and Nicolas Chopin. Sequential quasi Monte Carlo. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 77(3):509–579, 2015. doi: 10.1111/rssb.12104. URL https://doi.org/10.1111/rssb.12104.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Dejan Grubisic, Volker Seeker, Gabriel Synnaeve, Hugh Leather, John Mellor-Crummey, and Chris Cummins. Priority sampling of large language models for compilers. In *Proceedings of the 4th Workshop on Machine Learning and Systems*, pp. 91–97. ACM, 2024. doi: 10.1145/3642970.3655831. URL https://doi.org/10.1145/3642970.3655831.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z. URL https://doi.org/10.1038/s41586-025-09422-z.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper-round2.pdf.

John Hewitt, Christopher Manning, and Percy Liang. Truncation sampling as language model desmoothing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 3414–3427. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-emnlp.249. URL https://aclanthology.org/2022.findings-emnlp.249/.

Wassily Hoeffding. On the distribution of the number of successes in independent trials. *The Annals of Mathematical Statistics*, 27(3):713–721, 1956. doi: 10.1214/aoms/1177728178. URL https://doi.org/10.1214/aoms/1177728178.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://iclr.cc/virtual/2020/poster/1428.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, Kai Dang, Yang Fan, Yichang Zhang, An Yang, Rui Men, Fei Huang, Bo Zheng, Yibo Miao, Shanghaoran Quan, Yunlong Feng, Xingzhang Ren, Xuancheng Ren, Jingren Zhou, and Junyang Lin. Qwen2.5-Coder technical report. *arXiv preprint arXiv:2409.12186*, 2024. URL https://arxiv.org/abs/2409.12186.

Liwei Jiang, Yuanjun Chai, Margaret Li, Mickel Liu, Raymond Fok, Nouha Dziri, Yulia Tsvetkov, Maarten Sap, and Yejin Choi. Artificial hivemind: The open-ended homogeneity of language models (and beyond). In *Advances in Neural Information Processing Systems*, volume 38, pp. 90705–90774. Curran Associates, Inc., 2025. doi: 10.52202/085713-2732. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/754d5a526a5ee5a47220664a0eb92751-Paper-Datasets_and_Benchmarks_Track.pdf.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of RLHF on LLM generalisation and diversity. In *International Conference on Learning Representations*, pp. 20620–20653, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/5a68d05006d5b05dd9463dd9c0219db0-Paper-Conference.pdf.

Wouter Kool, Herke Van Hoof, and Max Welling. Stochastic beams and where to find them: The Gumbel-top-k trick for sampling sequences without replacement. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 3499–3508. PMLR, 2019. URL https://proceedings.mlr.press/v97/kool19a.html.

Suhas S Kowshik, Abhishek Divekar, and Vijit Malik. CorrSynth - a correlated sampling method for diverse dataset generation from LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 16076–16095. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.899. URL https://aclanthology.org/2024.emnlp-main.899/.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=FpaCL1MO2C.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=VD-AYtP0dve.

Sumith Kulal, Panupong Pasupat, Kartik Chandra, Mina Lee, Oded Padon, Alex Aiken, and Percy S Liang. SPoC: Search-based pseudocode to code. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/7298332f04ac004a0ca44cc69ecf6f6b-Paper.pdf.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626. ACM, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Hynek Kydlíček. Math-Verify: Math verification library, 2026. URL https://github.com/huggingface/Math-Verify. Software, version 0.9.0.

Jack Lanchantin, Angelica Chen, Shehzaad Dhuliawala, Ping Yu, Jason Weston, Sainbayar Sukhbaatar, and Ilia Kulikov. Diverse preference optimization. *arXiv preprint arXiv:2501.18101*, 2025. URL https://arxiv.org/abs/2501.18101.

Pierre L'Ecuyer. Randomized Quasi-Monte Carlo: An introduction for practitioners. In *Monte Carlo and Quasi-Monte Carlo Methods*, pp. 29–52. Springer International Publishing, 2018. doi: 10.1007/978-3-319-91436-7_2. URL https://doi.org/10.1007/978-3-319-91436-7_2.

Pierre L'Ecuyer, Christian Lécot, and Bruno Tuffin. A randomized Quasi-Monte Carlo simulation method for Markov chains. *Operations Research*, 56(4):958–975, 2008. doi: 10.1287/opre.1080.0556. URL https://doi.org/10.1287/opre.1080.0556.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 19274–19286. PMLR, 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 110–119. Association for Computational Linguistics, 2016. doi: 10.18653/v1/N16-1014. URL https://aclanthology.org/N16-1014/.

Michael Y. Li, Anthony Zhan, Kanishk Gandhi, Noah D. Goodman, and Emily B. Fox. QuasiMoTTo: Quasi-Monte Carlo test-time scaling. *arXiv preprint arXiv:2607.01179*, 2026. URL https://arxiv.org/abs/2607.01179.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson d'Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel J. Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu, and Oriol Vinyals. Competition-level code generation with AlphaCode. *Science*, 378(6624):1092–1097, 2022. doi: 10.1126/science.abq1158. URL https://doi.org/10.1126/science.abq1158.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, pp. 39578–39601, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/aca97732e30bcf1303bc22ac3924fd16-Paper-Conference.pdf.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 21558–21572. Curran Associates, Inc., 2023. doi: 10.52202/075280-0943. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/43e9d647ccd3e4b7b5baab53f0368686-Paper-Conference.pdf.

Chris J. Maddison, Daniel Tarlow, and Tom Minka. A* sampling. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/937debc749f041eb5700df7211ac795c-Paper.pdf.

M. D. McKay, R. J. Beckman, and W. J. Conover. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21 (2):239–245, 1979. doi: 10.1080/00401706.1979.10489755. URL https://doi.org/10.1080/00401706.1979.10489755.

Clara Meister, Afra Amini, Tim Vieira, and Ryan Cotterell. Conditional Poisson stochastic beam search. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 664–681. Association for Computational Linguistics, 2021a. doi: 10.18653/v1/2021.emnlp-main.52. URL https://aclanthology.org/2021.emnlp-main.52/.

Clara Meister, Martina Forster, and Ryan Cotterell. Determinantal beam search. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6551–6562. Association for Computational Linguistics, 2021b. doi: 10.18653/v1/2021.acl-long.512. URL https://aclanthology.org/2021.acl-long.512/.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. Locally typical sampling. *Transactions of the Association for Computational Linguistics*, 11:102–121, 2023. doi: 10.1162/tacl_a_00536. URL https://doi.org/10.1162/tacl_a_00536.

Meta. Llama 3.2: Model card. GitHub model card, 2024. URL https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD.md.

Meta AI. Llama 3.3 70B Instruct: Model card. Hugging Face model card, 2024. URL https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct.

Mistral AI. Mistral Small 3.2 24B Instruct: Model card. Hugging Face model card, 2025. URL https://huggingface.co/mistralai/Mistral-Small-3.2-24B-Instruct-2506.

Alistair Moffat, Radford M. Neal, and Ian H. Witten. Arithmetic coding revisited. *ACM Transactions on Information Systems*, 16(3):256–294, 1998. doi: 10.1145/290159.290162. URL https://doi.org/10.1145/290159.290162.

Minh Nguyen, Andrew Baker, Clement Neo, Allen Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. Turning up the heat: Min-p sampling for creative and coherent LLM outputs. In *International Conference on Learning Representations*, pp. 70333–70366, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/afa5f124e36bed5cc2125067005d43f5-Paper-Conference.pdf.

Art B. Owen. Monte Carlo variance of scrambled net quadrature. *SIAM Journal on Numerical Analysis*, 34(5):1884–1910, 1997. doi: 10.1137/S0036142994277468. URL https://doi.org/10.1137/S0036142994277468.

Vishakh Padmakumar and He He. Does writing with language models reduce content diversity? In *International Conference on Learning Representations*, pp. 642–669, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/02dec8877fb7c6aa9a79f81661baca7c-Paper-Conference.pdf.

Ji Won Park and Kyunghyun Cho. Efficient semantic uncertainty quantification in language models via diversity-steered sampling. In *Advances in Neural Information Processing Systems*, volume 38, pp. 32154–32183. Curran Associates, Inc., 2025. doi: 10.52202/085713-1080. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/2e49934cac6cb8604b0c67cfa0828718-Paper-Conference.pdf.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025. URL https://arxiv.org/abs/2412.15115.

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

Red Hat AI. Llama-3.3-70B-Instruct-FP8-dynamic: Model card. Hugging Face model repository, 2024. URL https://huggingface.co/RedHatAI/Llama-3.3-70B-Instruct-FP8-dynamic. FP8 weights and activations. Checkpoint revision f50dbad2c84590ca17dc51e207c34321b65ff14b.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

J. Rissanen and G. G. Langdon. Arithmetic coding. *IBM Journal of Research and Development*, 23(2):149–162, 1979. doi: 10.1147/rd.232.0149. URL https://doi.org/10.1147/rd.232.0149.

Chantal Shaib, Venkata S Govindarajan, Joe Barrow, Jiuding Sun, Alexa Siu, Byron C Wallace, and Ani Nenkova. Standardizing the measurement of text diversity: A tool and comparative analysis. In *Proceedings of The 14th International Joint Conference on Natural Language Processing and The 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics: System Demonstrations*, pp. 36–46. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.ijcnlp-demo.5. URL https://aclanthology.org/2025.ijcnlp-demo.5/.

Kensen Shi, David Bieber, and Charles Sutton. Incremental sampling without replacement for sequence models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 8785–8795. PMLR, 2020. URL https://proceedings.mlr.press/v119/shi20a.html.

Stewart Slocum, Asher Parker-Sartori, and Dylan Hadfield-Menell. Diverse preference learning for capabilities and alignment. In *International Conference on Learning Representations*, pp. 24760–24790, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/3df1eca840e82b11bbc33f68c773c38e-Paper-Conference.pdf.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, pp. 10131–10165, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/1b623663fd9b874366f3ce019fdfdd44-Paper-Conference.pdf.

I. M. Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112, 1967. doi: 10.1016/0041-5553(67)90144-9. URL https://doi.org/10.1016/0041-5553(67)90144-9.

Michael Stein. Large sample properties of simulations using Latin hypercube sampling. *Technometrics*, 29(2):143–151, 1987. doi: 10.1080/00401706.1987.10488205. URL https://doi.org/10.1080/00401706.1987.10488205.

Chloe H. Su, Zhe Ye, Samuel Tenka, Aidan Yang, Soonho Kong, and Udaya Ghai. Learning adaptive LLM decoding. *arXiv preprint arXiv:2603.09065*, 2026. URL https://arxiv.org/abs/2603.09065.

Runyan Tan, Shuang Wu, and Phillip Howard. p-less sampling: A robust hyperparameter-free approach for LLM decoding. In *International Conference on Learning Representations*, pp. 119506–119549, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/c2366657bff61d33c048c92972b19369-Paper-Conference.pdf.

Guy Tevet and Jonathan Berant. Evaluating the evaluation of diversity in natural language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 326–346. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.eacl-main.25. URL https://aclanthology.org/2021.eacl-main.25/.

Chandra Shekhara Kaushik Valmeekam, Krishna Narayanan, Dileep Kalathil, Jean-Francois Chamberland, and Srinivas Shakkottai. LLMZip: Lossless text compression using large language models. *arXiv preprint arXiv:2306.04050*, 2023. URL https://arxiv.org/abs/2306.04050.

Ashwin Vijayakumar, Michael Cogswell, Ramprasaath Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search for improved description of complex scenes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018. doi: 10.1609/aaai.v32i1.12340. URL https://doi.org/10.1609/aaai.v32i1.12340.

Luke Vilnis, Yury Zemlyanskiy, Patrick Murray, Alexandre Tachard Passos, and Sumit Sanghai. Arithmetic sampling: Parallel diverse decoding for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 35120–35136. PMLR, 2023. URL https://proceedings.mlr.press/v202/vilnis23a.html.

Christian Walder and Deep Tejas Karkhanis. Pass@K policy optimization: Solving harder reinforcement learning problems. In *Advances in Neural Information Processing Systems*, volume 38, pp. 152416–152445. Curran Associates, Inc., 2025. doi: 10.52202/085713-5096. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/df8a1a6306c3bcc89f6078759feadf62-Paper-Conference.pdf.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5776–5788. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Peter West and Christopher Potts. Base models beat aligned models at randomness and creativity. In *Conference on Language Modeling*, 2025. URL https://colm.cc/virtual/2025/poster/456.

Ian H. Witten, Radford M. Neal, and John G. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540, 1987. doi: 10.1145/214762.214771. URL https://doi.org/10.1145/214762.214771.

Justin Wong, Yury Orlovskiy, Alexander Shypula, Michael Luo, Sanjit Seshia, and Joseph Gonzalez. SimpleStrat: Diversifying language model generation with stratification. In *Advances in Neural Information Processing Systems*, volume 38, pp. 116906–116935. Curran Associates, Inc., 2025. doi: 10.52202/085713-3897. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/a99cbe298888dc87029faa80eb10d2b9-Paper-Conference.pdf.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for LLM problem-solving. In *International Conference on Learning Representations*, pp. 55824–55845, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/8c3caae2f725c8e2a55ecd600563d172-Paper-Conference.pdf.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-Math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024. URL https://arxiv.org/abs/2409.12122.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Jiayi Yuan, Hao Li, Xinheng Ding, Wenya Xie, Yu-Jhe Li, Wentian Zhao, Kun Wan, Jing Shi, Xia Hu, and Zirui Liu. Understanding and mitigating numerical sources of nondeterminism in LLM inference. In *Advances in Neural Information Processing Systems*, volume 38, pp. 169819–169851. Curran Associates, Inc., 2025. doi: 10.52202/085713-5653. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/f80094a824ba5912d4a2de169c404a40-Paper-Conference.pdf.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *Advances in Neural Information Processing Systems*, volume 38, pp. 57654–57689. Curran Associates, Inc., 2025. doi: 10.52202/085713-1933. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/537d5aa768c2d534016a4d06f87bc8fb-Paper-Conference.pdf.

Jiayi Zhang, Simon Yu, Derek Chong, Anthony Sicilia, Michael R. Tomz, Christopher D. Manning, and Weiyan Shi. Verbalized sampling: How to mitigate mode collapse and unlock LLM diversity. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=zloIrd77G5.

Yiming Zhang, Harshita Diddee, Susan Holm, Hanchen Liu, Xinyue Liu, Vinay Samuel, Barry Wang, and Daphne Ippolito. NoveltyBench: Evaluating language models for humanlike diversity. In *Conference on Language Modeling*, 2025. URL https://colm.cc/virtual/2025/poster/630.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023. URL https://arxiv.org/abs/2311.07911.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 1097–1100. ACM, 2018. doi: 10.1145/3209978.3210080. URL https://doi.org/10.1145/3209978.3210080.

Zachary Ziegler, Yuntian Deng, and Alexander Rush. Neural linguistic steganography. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1210–1215. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1115. URL https://aclanthology.org/D19-1115/.

A Proofs and Guarantees

Setting. Fix a prompt, model, temperature, truncation rule, and length limit. They define a distribution P over complete responses. All statements concern the mathematical sampling rules in exact arithmetic. Appendix B discusses float64. Write a response as $Y = F(Z, E)$, where Z is uniform on $[0, 1)$, E is independent auxiliary randomness (the fresh draws used after release), and F performs arithmetic decoding from Z until release and ordinary decoding afterwards. For any valid release rule, $F(Z, E) \sim P$ when Z is uniform.

Stratified sets. Split $[0, 1)$ into K slices S_j of width $1/K$. Draw Z_j uniformly on S_j and independent E_j, independently across j, set $Y_j = F(Z_j, E_j)$, and permute the K answers uniformly at random. The permutation does not change any set-level quantity.

A.1 Each answer keeps the model's distribution

Proposition 1. *Each output position has distribution P.*

Proof. For an event A, let $q_j = \Pr\{F(Z_j, E_j) \in A\}$. After the random permutation, a fixed position holds the answer of a uniformly chosen slice, so $\Pr\{Y \in A\} = K^{-1} \sum_j q_j$. A uniform slice with a uniform point inside it is a uniform point of $[0, 1)$, so this equals $\Pr\{F(Z, E) \in A\} = P(A)$. □

A.2 Coverage of any fixed event

Proposition 2. *For any fixed event A with $q = P(A)$, $\Pr\{\text{some } Y_j \in A\} = 1 - \prod_j (1 - q_j) \geq 1 - (1 - q)^K$, with equality if and only if all q_j are equal.*

Proof. Independence across slices gives the product. By the arithmetic–geometric mean inequality, $\prod_j (1 - q_j) \leq \left(K^{-1} \sum_j (1 - q_j)\right)^K = (1 - q)^K$, with equality exactly when the q_j coincide. The right side is the miss probability of K independent answers. This is classical stratification (Hoeffding, 1956, McKay et al., 1979, Stein, 1987). □

Taking A as the set of correct responses gives the pass@K guarantee of Equation 1. Several consequences follow by applying Proposition 2 to several events.

Distinct answers and fixed classes. For a fixed partition of responses into classes A_g (for example accepted answers, or meanings under a fixed labeling), the number of classes covered is $\sum_g \mathbf{1}\{\text{some } Y_j \in A_g\}$. Applying the proposition to each class and summing shows that its expectation never decreases. The same holds for classes intersected with a fixed quality threshold. Our semantic grouping depends on the sampled set and is therefore measured empirically.

Best answer under a fixed reward. For a reward $r(y) \in [0, 1]$, apply the proposition to $\{r > t\}$ for every t. Then $\max_j r(Y_j)$ stochastically dominates its independent counterpart, and $\mathbb{E} \max_j r(Y_j) \geq \mathbb{E} \max_j r(Y_j^{\text{iid}})$. A verifier must still identify the best answer (Gao et al., 2023).

Uniform subsets. If $k \leq K$ positions are chosen uniformly without replacement, the miss probability is a normalized elementary symmetric polynomial of the $1 - q_j$, which Maclaurin's inequality bounds by $(1 - q)^k$. The guarantee therefore also holds for subsets.

Variance of average quality. For a fixed statistic r with slice means μ_j and overall mean μ, the variance of the set average is smaller than under independent sampling by $K^{-2} \sum_j (\mu_j - \mu)^2$, the usual stratification identity.

Embedding diversity. For a fixed embedding ϕ with slice means m_j and mean m, the expected mean pairwise squared distance increases by $\frac{2}{K(K-1)} \sum_j \|m_j - m\|^2$. Expanding the squared distances, the norms agree by Proposition 1 and the cross terms factor by independence.

A.3 Proof of Theorem 1

Arithmetic decoding in slice S_j maintains the prefix interval $I(h) = [L, L + W)$. Conditional on the slice and on having generated h, the initial number Z is uniform on $I(h) \cap S_j$, because Z was uniform on S_j and h is generated exactly when $Z \in I(h)$.

Sufficiency. If $I(h) \subseteq S_j$, then $I(h) \cap S_j = I(h)$ and $(Z - L)/W$ is uniform on $[0, 1)$ given h and the slice. Arithmetic decoding of the continuation from this rescaled number therefore has the model's conditional distribution given h, and so does ordinary decoding with fresh draws. Replacing one by the other at the first prefix that satisfies containment changes no conditional law. The release time is a stopping time of the generated prefix, so the argument applies at each possible release prefix, and each answer has the same law as under always-on arithmetic decoding within its slice. Slices use independent numbers and independent fresh draws, so the joint law of the answer set is unchanged. Any fixed function of the set, including a verifier, a semantic grouping, or a diversity score, therefore has the same distribution.

Necessity. If $I(h) \not\subseteq S_j$, the rescaled number is uniform on a proper subinterval $J \subsetneq [0, 1)$, so its distribution function differs from the uniform one at some $u \in (0, 1)$. A next-token distribution with two tokens of probabilities u and $1 - u$ then selects the first token with a probability different from u under the retained number, but with probability exactly u under a fresh draw. No rule that uses only the slice and the prefix interval can release earlier for every model.

A.4 How long an answer stays coordinated

Proposition 3. *For a uniformly chosen output position and any $b \geq 0$, the probability that some prefix with surprisal at least $\log_2 K + b$ bits is reached while the answer still uses its number is at most $(1 - 1/K)2^{-b}$.*

Proof. Let $a = 2^{-b}/K$. On each decoding path take the first prefix whose interval width is at most a. These first-crossing prefixes form a prefix-free set, so their intervals have disjoint interiors. If the answer is still coordinated at or after such a prefix, it was coordinated at the first crossing, whose interval must then contain one of the $K - 1$ interior slice boundaries. Each boundary lies in at most one of these disjoint intervals, each of width at most a. Their total probability under a uniform initial number is at most $(K - 1)a = (1 - 1/K)2^{-b}$. □

With $K = 8$ and $b = 50$, an answer consumes more than 53 bits before release with probability below $7 \cdot 2^{-53} < 2^{-50}$. The bound needs no assumption on token probabilities and allows long runs of nearly deterministic tokens, which add almost no surprisal.

A.5 Why the lattice is a separate baseline

The lattice of Arithmetic Sampling (Figure 7a) uses $Z_j = (U + j/K) \bmod 1$ with one shared uniform U (Vilnis et al., 2023), a Cranley–Patterson rotation (Cranley & Patterson, 1976, L'Ecuyer, 2018). Each Z_j is uniform, so Proposition 1 holds, but the numbers are dependent and Proposition 2 can fail. For the event $A = \{z : \operatorname{frac}(Kz) < p\}$ all K numbers hit or miss A together, so the coverage is p, below the value $1 - (1 - p)^K$ of independent answers. A lattice analogue of release, singleton release, frees an answer once its prefix interval is no wider than the spacing $1/K$ (Figure 7b) and preserves each answer's distribution. We report the lattice with its original always-on decoding as the prior-work baseline.

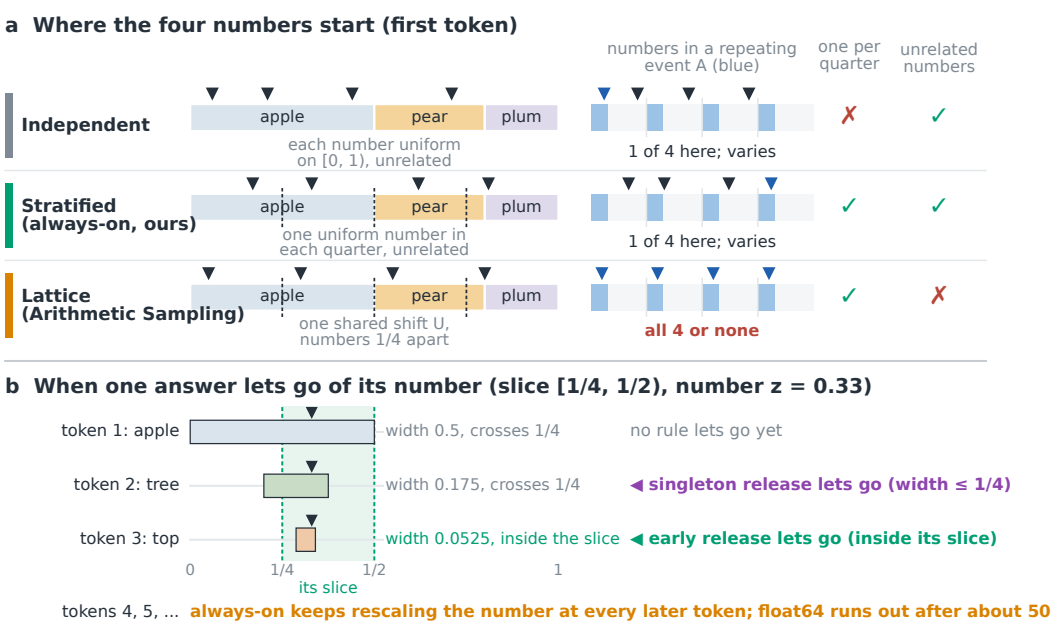

Figure 7: **How the coordination schemes differ.** Constructed example with $K = 4$ answers and the first-token distribution of Figure 1. (a) Where the numbers start. Independent sampling draws four unrelated uniform numbers, which can share a quarter. Stratified sampling, which always-on stratified sampling and early release both use, draws one uniform number in each quarter, independently. The lattice of Arithmetic Sampling shifts four equally spaced numbers by one shared uniform offset. In all three, each answer on its own follows the model. The right column shows an event A that repeats in every quarter, such as a rare opening phrase: independent and stratified numbers enter it separately, so Proposition 2 applies, while the four lattice numbers enter it all together or not at all, so the set contains it only with its single-answer probability. (b) When one answer lets go of its number, shown by its prefix interval in absolute position on $[0, 1)$. Always-on decoding never lets go. Singleton release, the release rule for the lattice, lets go once the prefix has probability at most $1/K$. For stratified numbers this would be too early, because the number is then uniform only on the part of the interval inside the slice. Early release lets go once the interval lies inside the slice, the earliest point at which fresh draws have the same law (Theorem 1).

B Implementation, Precision, and Cost

Sampler. All drawing rules are implemented as one vLLM logits processor (Kwon et al., 2023) selected per request. At each step it applies temperature and the truncation rule, renormalizes in float64, orders tokens by decreasing probability with ties broken by token id, and forms the cumulative distribution. An arithmetic request selects the token whose interval contains its number, updates the prefix interval $[L, L + W)$ in float64, and rescales its number. Early release compares the prefix interval with the request's slice after every step and hands the request back to vLLM's native sampler once the interval fits. Always-on stratified and lattice requests keep their numbers until the end. When truncation leaves fewer than 257 tokens, the processor sorts only those tokens. Slices are assigned to the K outputs by a seeded random permutation, and each output's offset inside its slice is an independent seeded uniform draw.

Checks. An independent rational-arithmetic oracle verifies on 90 finite probability trees that release preserves conditional laws, single-answer distributions, and subset coverage, without floating-point operations. CPU and CUDA tests compare the optimized processor with a reference implementation in 16 configurations, including ties, release, and batch reordering.

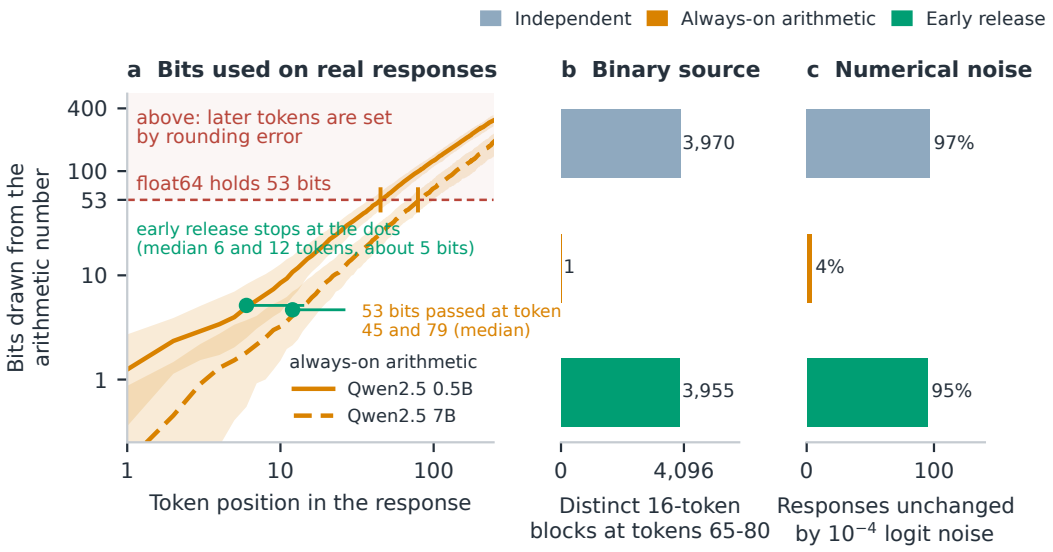

Figure 8: **Early release avoids exhausting the float64 number. (a)** Bits of u consumed by the prefix (its surprisal under the truncated distribution), for 800 responses per method and model on all 100 curated NoveltyBench requests (temperature 1, min-p 0.05). Lines are medians, bands interquartile ranges. Always-on arithmetic sampling passes 53 bits after a median of 44–79 tokens. Early release stops after 6–12 tokens (dots, whiskers to the 90th percentile). **(b)** A synthetic source with two equally likely tokens: distinct 16-token blocks at positions 65–80 among 4,096 responses. **(c)** Responses unchanged when Gaussian noise of standard deviation 10^{-4} is added to every logit (Qwen2.5 0.5B, 768 responses).

B.1 The float64 diagnostic

Figure 8a uses a standalone sampler that reproduces the processor's float64 update, probability ordering, and clamp, and adds per-token logging. For all 100 curated NoveltyBench requests, and separately for 128 longer WildChat requests from NoveltyBench, it draws eight answers with each rule at temperature 1 and min-p 0.05, for up to 768 tokens. Always-on and released requests of the same set use the same slices and offsets. For each token we record its surprisal under the truncated distribution. Table 2 summarizes both models and both prompt sets. The late-token argmax rate stays close to that of independent sampling: the exhausted number does not collapse on real text, because rounding in the rescaling step acts as a deterministic pseudo-random input. The output is then a function of rounding error instead of the random draw, much as numerical precision changes the outputs of ordinary inference (Yuan et al., 2025), and it can collapse on sources such as the binary example in Figure 8b, where every rescaling is an exact doubling that shifts out one bit.

Sensitivity to numerical noise. Figure 8c runs the same sampler twice on 32 NoveltyBench prompts with Qwen2.5 0.5B in float32 on CPU, for up to 256 tokens and eight answers per set. Both runs use identical slices, offsets, and fresh-draw streams. The second run adds independent Gaussian noise with standard deviation 10^{-4} to every logit from a separate generator, which leaves the random inputs untouched. We compare the generated texts response by response. Identical outputs occur for 96.9% of independent responses, 94.9% of early-release responses, and 3.9% of always-on arithmetic responses. Where outputs differ, the median first differing token is 145, 83, and 17, respectively.

B.2 Generation time

Every complete run of the benchmark campaign times each generation call for a block of prompts and one combination of drawing rule, truncation rule, and temperature. The order of these calls is randomized within each block, so the total time of each drawing rule over a run compares the rules on identical prompts and settings. Table 3 reports, for each task, the geometric mean over runs of each rule's time per answer relative to independent sampling. Times exclude model loading. Early release

Table 2: **Float64 use on real responses.** Medians over responses unless stated. Surprisal is measured under the truncated distribution.

Quantity	Curated (100) 0.5B	Curated (100) 7B	WildChat (128) 0.5B	WildChat (128) 7B
Response length (tokens)	120	77	356	512
Response surprisal (bits)	156	44	424	415
Always-on: token where 53 bits are passed	45	79	44	73
Always-on: responses that pass 53 bits (%)	84.6	47.1	93.1	94.7
Always-on: mean share of tokens after 53 bits (%)	54.4	27.8	76.4	76.1
Early release: release token	6	12	5	10
Early release: release token, 90th percentile	14	26	12	21
Early release: bits used, median	5.1	4.7	5.2	5.0
Early release: bits used, maximum	14.7	14.6	15.1	17.3
Late-token argmax rate, independent (%)	69.4	78.5	69.6	77.3
Late-token argmax rate, always-on (%)	68.6	78.3	69.2	77.2
Late-token argmax rate, early release (%)	69.1	78.6	69.4	77.5

is 14% faster than always-on stratified sampling in the runs that contain both, and the two produce the same answer-set distribution in exact arithmetic. The overhead of early release is smallest for long open-ended responses, where coordination occupies a small fraction of the tokens.

Table 3: **Relative generation time on the complete benchmark runs.** Time per answer relative to independent sampling in the same run (geometric mean over runs). Lower is better. The math runs did not include always-on strata.

Task (runs)	Independent	Early release	Always-on strata	Lattice
CoverageQA (9)	1.00	1.09	1.20	1.21
TriviaQA (7)	1.00	1.22	1.39	1.39
NoveltyBench (3)	1.00	1.02	1.30	1.28
IFEval (2)	1.00	1.02	1.44	1.43
MATH-500 / GSM8K (4)	1.00	1.12	–	1.52
All runs (25)	1.00	1.12	1.30	1.33

B.3 When to release, and how many bits the number needs

Early release fixes two choices: coordination ends at containment, and the number is a float64. We vary each on Qwen2.5 0.5B with min-p 0.05 at temperature 1 and eight answers, on all 145 CoverageQA questions (three seeds), all 11,313 TriviaQA questions, and all 1,319 GSM8K problems. Answers in the same position of a set share their slice, offset, and native seed across all rules, so the rules differ only in when and how the number is used.

Release time. *Release after n tokens* hands an answer to the native sampler after exactly n coordinated tokens, whether or not its interval fits its slice. Each answer still follows the model, because the release time depends only on the prefix, but the set loses part of the joint law whenever release comes before containment (Theorem 1). *Early release $+d$* keeps the number for d tokens after containment, which changes nothing in exact arithmetic and only spends bits.

Precision. We store the number with P significant bits and rescale it with correctly rounded arithmetic (MPFR) against the exact float64 cumulative probabilities, so only the precision of the number changes. $P = 53$ is float64, and $P = 2048$ is exact for every answer we generate. Figure 6 and Table 4 give the results.

Coordination is needed only briefly. On CoverageQA a single coordinated token already yields a 16% gain in distinct accepted answers, against 17% for early release and for always-on sampling. On TriviaQA the relative gain in pass@8 is 13% after one token, 16% after eight, 16% for early release, and 17% for always-on sampling, with overlapping intervals. Early release reaches this level after

a median of 10 tokens and 5.0 bits without a hand-chosen n, and keeping the number longer adds nothing. No rule changes TriviaQA pass@1 by more than 0.03 points.

Losing bits costs exactness and leaves accuracy intact. On GSM8K, always-on sampling chooses 37% of all tokens after float64 has run out, and 91% when the number keeps 8 bits, yet no rule moves pass@1 (independent sampling: 39.0%) or pass@8 beyond sampling noise: 29 of the 30 intervals contain zero, and the remaining one is an increase (pass@8 with 128 bits). Rounding error acts as a deterministic pseudo-random input, so on this task exhaustion damages the stated distribution and reproducibility (Appendix B.1) and leaves single-answer accuracy intact. Early release chooses no token with an exhausted float64 number, and 0.7% of tokens when the number keeps only 8 bits, because it lets go after a median of 4.8 bits.

Table 4: **Release time and precision, every rule.** Qwen2.5 0.5B, min-p 0.05, temperature 1, eight answers. The first row gives the value for independent sampling, and the other rows give the change from it (distinct accepted answers for CoverageQA, percentage points otherwise). Bits at end is the median surprisal consumed when an answer stops using its number (release or end of answer), on GSM8K where available and TriviaQA otherwise.

Drawing rule	CoverageQA distinct	TriviaQA pass@8	TriviaQA pass@1	GSM8K pass@1	GSM8K pass@8	Bits at end
Independent	1.54	5.0	1.4	39.0	75.3	–
Early release (ours)	+0.26	+0.80	+0.02	+0.15	+0.61	5
Release after 1 token	+0.25	+0.67	+0.01	–	–	0
Release after 2 tokens	+0.26	+0.61	+0.00	–	–	1
Release after 4 tokens	+0.26	+0.74	+0.02	+0.79	-0.91	1
Release after 8 tokens	+0.26	+0.81	+0.02	–	–	3
Release after 16 tokens	+0.26	+0.84	+0.02	+0.90	+0.68	4
Release after 32 tokens	+0.26	–	–	–	–	–
Release after 64 tokens	–	–	–	+0.45	+0.30	22
Early release +8 tokens	+0.26	–	–	–	–	–
Early release +32 tokens	+0.26	–	–	-0.53	-0.23	18
Early release +128 tokens	–	–	–	+0.00	-0.08	40
Early release, 8 bits	+0.25	+0.76	+0.01	+0.14	+0.76	5
Early release, 12 bits	+0.26	–	–	–	–	–
Early release, 16 bits	+0.26	+0.80	+0.02	+0.29	+0.76	5
Always-on, float64	+0.26	+0.85	+0.03	-0.14	+0.08	65
Always-on, 8 bits	+0.26	+0.83	+0.02	-0.15	-0.45	65
Always-on, 12 bits	+0.26	–	–	–	–	–
Always-on, 16 bits	+0.26	+0.85	+0.03	-0.19	+1.29	65
Always-on, 24 bits	+0.26	+0.85	+0.03	+0.38	+1.21	65
Always-on, 53 bits (emulated)	+0.26	–	–	-0.09	+1.36	65
Always-on, 128 bits	–	–	–	+0.41	+2.43	65
Always-on, exact (2048 bits)	+0.26	+0.85	+0.03	+0.63	+0.15	65

C Evaluation Details

Inference. All CUDA experiments use vLLM 0.20 in bfloat16 on H100 GPUs, with prefix caching and at most 128 to 256 concurrent sequences. Llama 3.3 70B uses the FP8 checkpoint with tensor parallelism over two GPUs. Only the answer-budget study (Appendix D.1) runs elsewhere: SmolLM2 135M and 360M and Llama 3.2 1B with Hugging Face Transformers in float16 on the MPS backend of an Apple M4 Pro, whose sampler computes the truncated distribution on the CPU with the same float64 arithmetic, and independent draws there also use float64 inverse-CDF sampling. Each model uses its chat template with default system prompt, and checkpoint revisions are pinned. Within a comparison, all drawing rules share the engine, prompts, and token limits, and their order is randomized within blocks of prompts. Seeds for fresh draws are derived from the model, prompt, replicate, and output position, so independent sampling and the post-release draws of early release use the same seed schedule.

CoverageQA. We use all 145 questions of the released repository (Wong et al., 2025) with a 64-token limit. An answer is accepted when its whole string matches an accepted alias after case folding and whitespace trimming. Aliases of one entity count once. We report the number of distinct accepted entities per set and the fraction of accepted answers.

TriviaQA. We use all 11,313 questions of the unfiltered validation split without context (Joshi et al., 2017). The prompt is "Answer the question with only the answer. Do not include an explanation."

followed by the question, with a 64-token limit. An answer is correct when its whole normalized string, under the official TriviaQA normalization, equals a normalized alias.

NoveltyBench. We use all 100 requests of the curated split (Zhang et al., 2025) with a 1,024-token limit. Mistral Small 3.2 24B (Mistral AI, 2025) and Qwen3.5 27B (Qwen Team, 2026) independently rate each complete response from 0 to 4 for correctness, relevance, coherence, and usefulness, without seeing the method. A response is approved when both ratings are at least 3. Approved responses are then grouped by meaning with NoveltyBench's released similarity classifier and greedy partition, a learned notion of equivalent meaning related to semantic entropy (Kuhn et al., 2023, Farquhar et al., 2024). Quality-approved meanings are the number of resulting groups, and an empty approved set has zero meanings. Diversity measures on complete responses use the pinned `all-MiniLM-L6-v2` encoder (Reimers & Gurevych, 2019, Wang et al., 2020) over non-overlapping chunks, the Vendi score (Friedman & Dieng, 2023), mean pairwise cosine distance, distinct n-grams (Li et al., 2016), pairwise n-gram distances (Zhu et al., 2018), and unique strings (Tevet & Berant, 2021, Shaib et al., 2025).

Reasoning and instruction following. MATH-500 (Lightman et al., 2024, Hendrycks et al., 2021) and the full GSM8K test set (Cobbe et al., 2021) use the Qwen2.5-Math system prompt "Please reason step by step, and put your final answer within `\boxed{}`" with a 4,096-token context. Answers are checked with Math-Verify 0.9 (Kydlíček, 2026), with GSM8K's `####` form as a fallback. IFEval uses all 541 prompts (Zhou et al., 2023), a 2,048-token limit, and the official evaluator. We report prompt-level strict accuracy.

Code. HumanEval+ (all 164 problems) and MBPP+ (all 378 problems) come from EvalPlus (Chen et al., 2021, Austin et al., 2021, Liu et al., 2023). We use EvalPlus's chat format: the instruction "Please provide a self-contained Python script that solves the following problem in a markdown code block" followed by the problem, and a response that begins with EvalPlus's fixed preamble and an opening Python code block, with a 1,024-token limit. The completion is extracted with EvalPlus's sanitizer and run against both the original and the extended EvalPlus tests. We report the extended (+) tests. The eight models are SmolLM2 360M and 1.7B, Qwen2.5 0.5B, 1.5B, and 7B, Qwen2.5-Coder 1.5B and 7B (Hui et al., 2024), and Llama 3.2 1B, all Instruct, at temperature 1 with the four truncation rules.

Reasoning models. DeepSeek-R1-Distill-Qwen 1.5B and 7B (Guo et al., 2025) and Qwen3 1.7B in thinking mode (Yang et al., 2025) answer all MATH-500 problems and the 60 AIME 2024 and 2025 problems, with the instruction "Please reason step by step, and put your final answer within `\boxed{}`" appended to the problem, as recommended for these models, and top-p 0.95 at temperature 0.6. The limit is 16,384 tokens for MATH-500 and 30,000 for AIME. The answer is the last boxed expression after the end of the reasoning trace, and a trace cut off by the limit counts as wrong. Long traces make the engine preempt requests and resume them later. The processor then restores a coordinated request's arithmetic state, and for the few early-release answers whose state could not be restored (at most 1% per model) generation continues with native sampling, which can only dilute a difference between the methods.

Statistics. All differences are paired by prompt. For each prompt we average over seeds and settings with fixed equal weights, then draw 10,000 to 20,000 bootstrap resamples of prompts that keep all methods, models, and settings of a prompt together, and report percentile intervals. Per-setting intervals are Bonferroni-adjusted over the settings shown in the same figure or table. The equivalence test for individual acceptance on CoverageQA uses a 90% interval and a margin of one percentage point fixed before generation. For NoveltyBench, the primary comparison and its 97.5% intervals were declared before the responses were generated.

D Detailed Results

D.1 CoverageQA

Temperature grid. Table 5 averages the complete grid of Figure 5 for all four drawing rules, and Table 6 lists every setting of every model. The design was fixed before generation and uses three

fresh seeds per setting. Always-on strata and early release estimate the same answer-set distribution, so their difference reflects sampling noise. The grids of Qwen2.5 0.5B and 7B and Llama 3.3 70B come from the original campaign, and the other four models were added later with the same design, prompts, and engine. The pooled change in the share of accepted answers is -0.12 points (95% interval [-0.36, +0.12]), inside the prespecified equivalence margin of one point. Per model it ranges from -0.6 points (Llama 3.3 70B) to +0.6 points (Llama 3.2 1B).

The two negative SmolLM2 cells. Only two of the 112 cells are clearly negative: SmolLM2 360M and 1.7B with top-p 0.9 at temperature 2 (-30% and -17% in Figure 9). They are not failures of early release. At this setting both models almost never produce a valid answer: 0.8% and 1.2% of independent answers are accepted. Summed over the 435 sets of a cell (145 questions, three seeds), independent sampling finds 27 and 40 distinct accepted answers and early release 19 and 33, so a difference of a few answers becomes a large relative change. Neither difference is significant even before adjustment (95% intervals $[-0.048, +0.011]$ and $[-0.057, +0.023]$ distinct answers per set). Always-on stratified sampling, which has the same distribution as early release in exact arithmetic, finds 30 distinct answers in the 360M cell, more than independent sampling, which confirms that the gap is sampling noise at a near-empty cell. With min-p or top-k at the same temperature, where the models still answer, early release gains 3 to 11%.

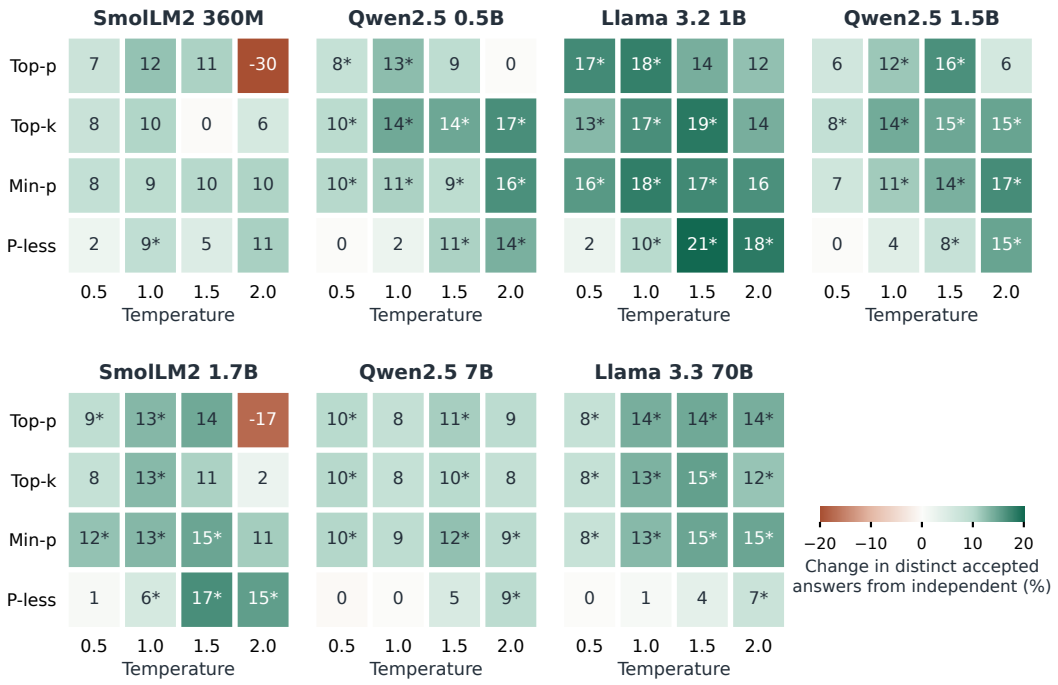

Figure 9: **CoverageQA grid for all seven models.** Same quantities as Figure 5. The two negative cells, SmolLM2 360M and 1.7B with top-p at temperature 2, are not significant: there only about 1% of answers are accepted, so a cell holds a few dozen distinct accepted answers over all 435 sets and a difference of a few answers is a large relative change (see the text above and Table 6).

Table 5: **CoverageQA, complete temperature grid.** Means over seven models, four truncation rules, four temperatures, three seeds, and all 145 questions. Bold marks the highest displayed value.

Method	Distinct accepted	Individual accepted (%)
Independent	1.371	**44.59**
Early release	1.528	44.47
Always-on strata	1.529	44.47
Lattice	**1.564**	44.38

Table 6: **Every CoverageQA setting.** Distinct accepted answers per eight-answer set with independent sampling and early release, averaged over three seeds and all 145 questions. An asterisk marks an early-release gain whose interval excludes zero after Bonferroni adjustment over all 112 settings. Values below 0.1 (SmolLM2 360M and 1.7B and Llama 3.2 1B with top-p at temperature 2) mark settings where the model almost never gives a valid answer, so their differences are noise. The per-setting values of the controls are in the released compact data.

		$\tau = 0.5$		$\tau = 1.0$		$\tau = 1.5$		$\tau = 2.0$	
Model	Rule	Ind.	Early	Ind.	Early	Ind.	Early	Ind.	Early
SmolLM2 360M	Top-p	1.27	1.36	1.58	1.77	0.62	0.69	0.06	0.04
	Top-k	1.33	1.43	1.50	1.66	1.09	1.09	0.61	0.65
	Min-p	1.19	1.29	1.62	1.77	1.27	1.40	0.74	0.82
	P-less	0.65	0.66	1.12	1.22*	1.76	1.84	1.17	1.30
Qwen2.5 0.5B	Top-p	1.08	1.16*	1.70	1.92*	1.45	1.58	0.16	0.16
	Top-k	1.19	1.31*	1.72	1.97*	1.76	2.02*	1.42	1.66*
	Min-p	1.09	1.20*	1.62	1.79*	1.94	2.12*	1.77	2.06*
	P-less	0.62	0.62	0.83	0.85	1.40	1.55*	1.89	2.15*
Llama 3.2 1B	Top-p	1.27	1.48*	1.88	2.21*	1.23	1.40	0.09	0.11
	Top-k	1.40	1.58*	1.86	2.17*	1.54	1.82*	1.01	1.15
	Min-p	1.22	1.41*	1.89	2.23*	1.90	2.22*	1.53	1.78
	P-less	0.61	0.62	0.94	1.04*	1.60	1.94*	1.73	2.05*
Qwen2.5 1.5B	Top-p	1.00	1.06	1.57	1.75*	1.86	2.16*	0.61	0.65
	Top-k	1.13	1.22*	1.68	1.91*	1.89	2.16*	1.62	1.86*
	Min-p	1.00	1.07	1.59	1.77*	1.89	2.15*	1.94	2.26*
	P-less	0.57	0.57	0.77	0.80	1.07	1.15*	1.67	1.92*
SmolLM2 1.7B	Top-p	1.33	1.45*	2.04	2.31*	1.08	1.24	0.09	0.08
	Top-k	1.52	1.65	2.01	2.27*	1.75	1.94	1.11	1.14
	Min-p	1.36	1.52*	2.01	2.27*	2.09	2.39*	1.63	1.81
	P-less	0.70	0.71	1.21	1.28*	1.78	2.08*	1.98	2.29*
Qwen2.5 7B	Top-p	0.97	1.07*	1.37	1.48	1.64	1.82*	1.33	1.46
	Top-k	1.09	1.20*	1.51	1.63	1.83	2.01*	2.05	2.22
	Min-p	1.01	1.11*	1.39	1.51	1.68	1.88*	1.96	2.15*
	P-less	0.70	0.70	0.78	0.78	0.97	1.02	1.34	1.47*
Llama 3.3 70B	Top-p	1.03	1.12*	1.50	1.71*	2.02	2.31*	2.29	2.61*
	Top-k	1.17	1.26*	1.71	1.94*	2.22	2.55*	2.55	2.86*
	Min-p	1.09	1.17*	1.54	1.74*	2.03	2.33*	2.43	2.78*
	P-less	0.75	0.75	0.88	0.89	0.97	1.00	1.27	1.37*

Answer budgets. Figure 10 generates new sets of 2, 4, 8, 16, and 32 answers, each averaged over the four truncation rules at temperature 1, for SmolLM2 360M, Llama 3.2 1B, and SmolLM2 135M. The gain interval is adjusted over these three models. SmolLM2 135M produces almost no accepted answers in the required format at any budget (Table 7) and is omitted from the figure.

Table 7: **Answer-budget study, averaged over budgets and truncation rules.** Bold marks the highest value within a model.

Model	Rule	Accepted rate (%)	Any accepted rate (%)	Distinct entities	Effective entities	Simpson entities	Entropy (nats)
SmolLM2 135M	Independent	**0.15**	1.52	**0.02**	**0.02**	**0.02**	**0.00**
	Early release	0.14	**1.62**	**0.02**	**0.02**	**0.02**	**0.00**
	Lattice	0.12	1.31	0.01	0.01	0.01	**0.00**
SmolLM2 360M	Independent	28.25	58.93	1.55	1.39	1.28	0.37
	Early release	29.17	61.66	1.83	1.67	1.55	0.44
	Lattice	**29.63**	**62.00**	**1.90**	**1.74**	**1.62**	**0.47**
Llama 3.2 1B	Independent	**40.22**	62.48	1.87	1.65	1.51	0.44
	Early release	39.47	**62.66**	2.11	1.88	1.72	0.51
	Lattice	39.33	62.14	**2.16**	**1.92**	**1.76**	**0.52**

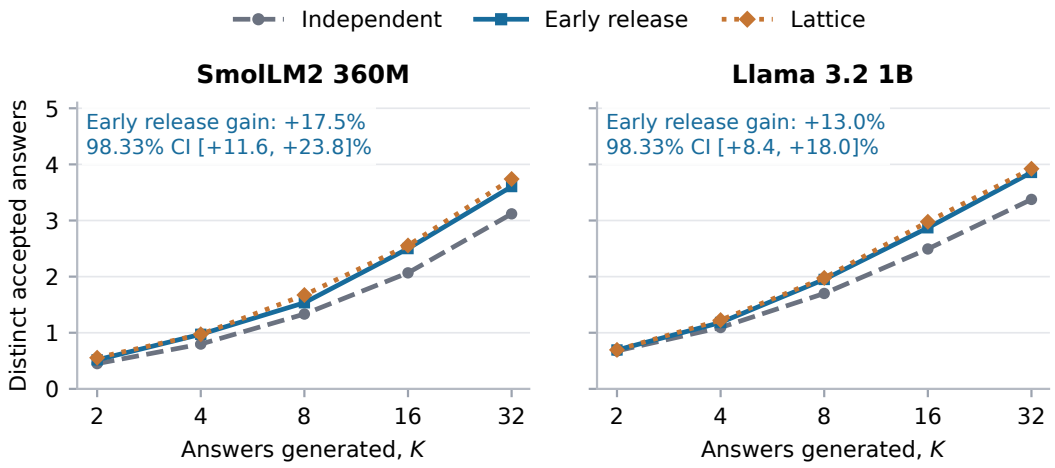

Figure 10: **Coverage at five answer budgets.** Each budget is generated separately. The early-release gain averages the five budgets and four truncation rules.

D.2 TRIVIAQA

Figure 11 and Table 8 give every model and truncation rule, with all four drawing rules.

Table 8: **Complete TriviaQA validation set.** Pass@8 and pass@1 (%) over all 11,313 questions at temperature 1. The interval is for the pass@8 difference between early release and independent sampling, Bonferroni-adjusted over the 28 settings.

		Pass@8				Pass@1				
Model	Rule	Ind.	Early	Strata	Lattice	Ind.	Early	Strata	Lattice	Early − ind. (pass@8)
SmolLM2 360M	Top-p	8.91	9.99	9.91	**10.16**	**2.64**	2.61	2.60	2.62	[+0.36, +1.84]
	Top-k	8.57	9.84	9.85	**10.11**	2.41	2.44	2.44	**2.45**	[+0.55, +1.99]
	Min-p	9.37	10.10	9.99	**10.50**	3.04	3.00	2.98	**3.05**	[+0.07, +1.42]
	P-less	6.36	6.84	6.82	**6.94**	**3.77**	3.73	3.73	3.73	[+0.20, +0.79]
Qwen2.5 0.5B	Top-p	5.50	5.93	5.94	**6.08**	**1.35**	1.31	1.32	1.32	[-0.19, +1.08]
	Top-k	5.65	5.81	5.98	**6.05**	**1.29**	1.22	1.25	1.24	[-0.54, +0.79]
	Min-p	5.34	5.75	5.77	**5.98**	**1.43**	1.39	1.40	1.38	[-0.14, +0.97]
	P-less	2.28	2.35	2.36	**2.40**	**1.33**	**1.33**	**1.33**	1.32	[-0.06, +0.21]
Llama 3.2 1B	Top-p	34.60	37.36	37.15	**37.95**	17.58	**17.65**	17.63	17.59	[+1.80, +3.73]
	Top-k	35.00	37.26	37.03	**37.94**	**16.76**	16.69	16.65	16.74	[+1.26, +3.20]
	Min-p	36.15	38.23	38.20	**39.28**	**19.13**	19.09	19.08	**19.13**	[+1.15, +3.01]
	P-less	30.35	31.15	31.14	**31.39**	22.77	**22.81**	**22.81**	22.78	[+0.42, +1.22]
Qwen2.5 1.5B	Top-p	36.90	38.42	38.52	**39.30**	22.73	22.72	22.74	**22.82**	[+0.68, +2.34]
	Top-k	37.13	38.84	38.88	**39.77**	21.64	21.64	21.66	**21.70**	[+0.79, +2.60]
	Min-p	37.37	39.18	39.08	**39.57**	23.32	**23.42**	23.40	**23.42**	[+1.03, +2.60]
	P-less	29.83	**30.34**	30.33	30.28	**25.97**	25.96	25.96	25.96	[+0.27, +0.73]
SmolLM2 1.7B	Top-p	36.75	**40.07**	**40.07**	39.98	18.46	18.59	**18.61**	18.45	[+2.32, +4.35]
	Top-k	36.64	39.13	39.19	**39.64**	**17.23**	17.13	17.14	17.22	[+1.54, +3.43]
	Min-p	37.31	40.35	40.22	**40.71**	19.56	**19.67**	**19.67**	19.59	[+2.17, +4.00]
	P-less	29.42	30.21	**30.24**	30.23	23.21	23.33	**23.35**	23.30	[+0.50, +1.14]
Qwen2.5 7B	Top-p	51.19	52.63	52.66	**52.72**	42.04	**42.09**	42.08	42.02	[+0.87, +2.05]
	Top-k	52.00	53.41	53.39	**53.63**	41.25	41.21	41.21	**41.29**	[+0.72, +2.08]
	Min-p	51.51	**53.18**	53.12	53.17	42.32	**42.41**	42.40	42.37	[+1.08, +2.27]
	P-less	45.43	**45.47**	**45.47**	45.46	**43.75**	43.73	43.74	43.73	[-0.06, +0.16]
Llama 3.3 70B (FP8)	Top-p	76.20	**76.61**	76.58	76.55	72.18	**72.23**	72.22	72.20	[+0.15, +0.69]
	Top-k	76.86	**77.61**	77.53	77.51	71.82	**71.92**	71.91	71.89	[+0.34, +1.19]
	Min-p	76.50	76.93	76.91	**76.95**	72.10	**72.16**	72.15	72.14	[+0.10, +0.75]
	P-less	**72.92**	**72.92**	**72.92**	**72.92**	72.64	**72.65**	**72.65**	**72.65**	[+0.00, +0.04]

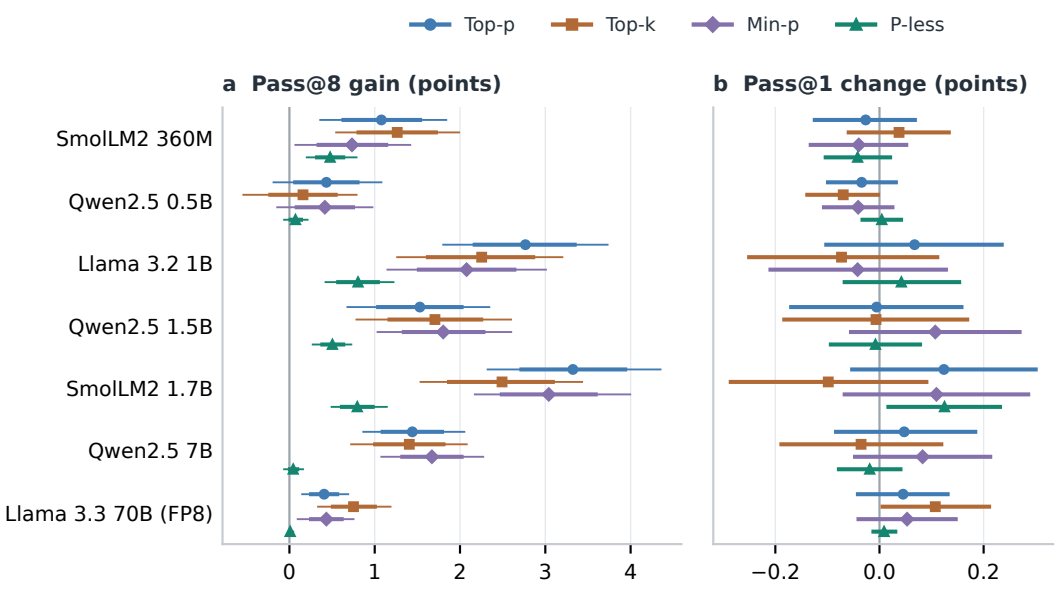

Figure 11: **Higher pass@8 on the complete TriviaQA validation set.** Each point is the paired pass@8 difference between early release and independent sampling on all 11,313 questions for one model and truncation rule, at temperature 1 with eight answers. Thick bars are 95% bootstrap intervals and thin bars are Bonferroni-adjusted intervals over the 28 settings. The right panel shows the same comparison for single-answer accuracy, which the method leaves unchanged by construction.

D.3 NoveltyBench

Table 9: **Quality and meanings on all 100 curated requests.** Means over three models, four truncation rules, and two temperatures. Quality-approved meanings count groups among responses that both judges approve. Passing is the share of individual responses both judges approve. Any passing is the share of sets with at least one such response. Best rating is the highest, within a set, of the smaller of the two ratings. Bold marks the highest displayed value.

Method	Qualified groups	Passing (%)	Any passing (%)	Best rating
Independent	1.836	**59.54**	77.25	3.381
Early release	1.902	59.09	77.38	**3.391**
Always-on strata	1.903	59.33	**77.67**	3.385
Lattice	**1.906**	59.19	76.79	3.371

Table 10: **Additional measurements on the same grid.** Effective and Simpson counts reward an even spread over approved meanings. Length and cap rates are descriptive.

Measure	Independent	Early release	Strata	Lattice
Groups before quality filtering	4.350	4.461	4.449	**4.470**
Effective qualified groups	1.698	1.759	**1.761**	**1.761**
Simpson qualified groups	1.611	1.668	**1.670**	1.669
Mean conservative rating	**2.898**	2.891	2.893	2.893
Mean output tokens	202.5	203.2	203.7	201.3
Responses reaching the cap (%)	4.69	4.67	4.62	4.43

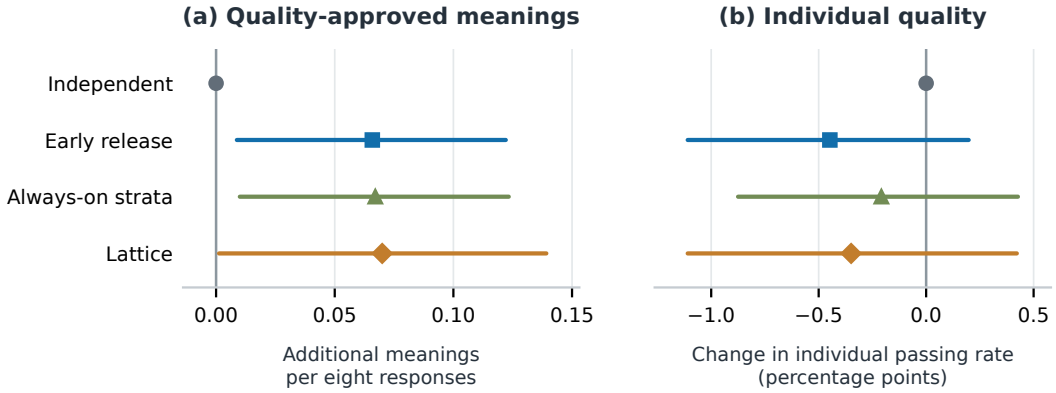

Figure 12: **More quality-approved meanings with unchanged individual quality.** Left: additional distinct meanings among responses that both judges approve, per set of eight. Right: change in the percentage of individual responses that both judges approve. Means cover all 100 curated NoveltyBench requests, three models, four truncation rules, and two temperatures. Bars are paired 97.5% bootstrap intervals. Always-on stratified sampling and the lattice are controls.

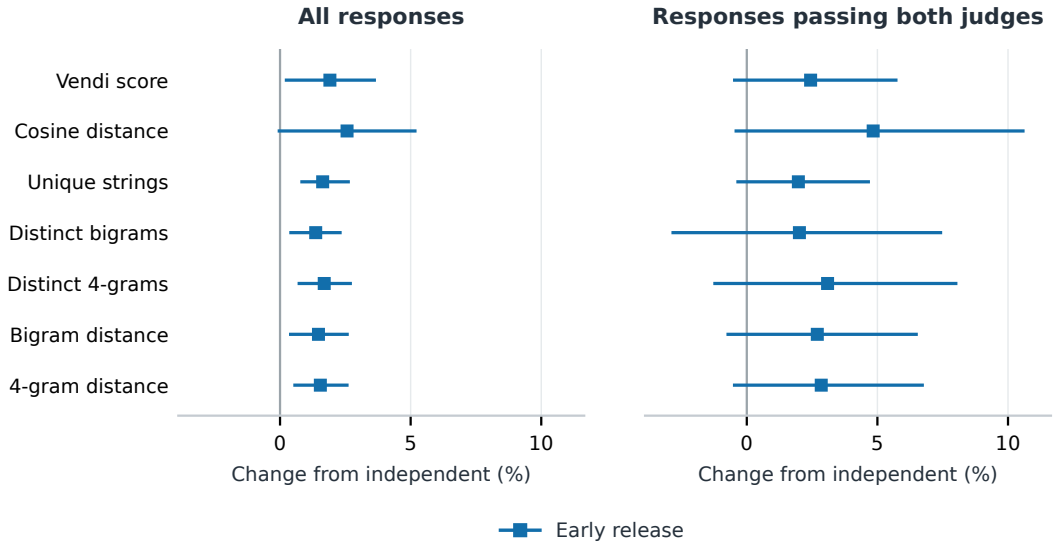

Figure 13: **Seven further diversity measures.** Relative change from independent sampling, on all responses (left) and on approved responses (right). Intervals are Bonferroni-adjusted over the declared family of comparisons. Six of the seven unfiltered measures improve after adjustment. On approved responses all seven point estimates are positive and the adjusted intervals include zero.

Table 11: **Quality-approved meanings for every setting.** Per model, truncation rule, and temperature. Intervals are Bonferroni-adjusted over all settings.

Qwen2.5 0.5B

Filter	τ	Independent	Early release	Strata	Lattice	Early release − independent
Top-p	1.0	**1.140**	**1.140**	1.110	1.080	[-0.316, +0.336]
Top-p	1.5	0.060	**0.080**	0.050	0.040	[-0.080, +0.130]
Top-k	1.0	0.980	1.040	1.010	**1.070**	[-0.320, +0.436]
Top-k	1.5	**0.340**	0.280	0.270	0.230	[-0.300, +0.220]
Min-p	1.0	**1.380**	1.340	1.270	1.260	[-0.400, +0.320]
Min-p	1.5	0.830	0.690	**0.860**	0.730	[-0.446, +0.150]
P-less	1.0	**1.020**	0.880	0.950	0.990	[-0.430, +0.120]
P-less	1.5	1.430	**1.460**	1.370	1.310	[-0.290, +0.360]

Qwen2.5 7B

Filter	τ	Independent	Early release	Strata	Lattice	Early release − independent
Top-p	1.0	2.830	2.930	**2.980**	2.810	[-0.280, +0.490]
Top-p	1.5	2.150	2.060	2.090	**2.310**	[-0.500, +0.360]
Top-k	1.0	3.150	**3.190**	3.130	**3.190**	[-0.410, +0.500]
Top-k	1.5	3.270	3.330	3.480	**3.580**	[-0.410, +0.530]
Min-p	1.0	2.920	2.950	2.830	**2.970**	[-0.356, +0.390]
Min-p	1.5	3.340	3.620	**3.770**	3.680	[-0.140, +0.706]
P-less	1.0	1.330	1.510	**1.530**	1.520	[+0.010, +0.390]
P-less	1.5	2.060	**2.170**	2.160	2.020	[-0.190, +0.390]

Llama 3.3 70B (FP8)

Filter	τ	Independent	Early release	Strata	Lattice	Early release − independent
Top-p	1.0	1.900	**2.100**	2.060	1.980	[-0.096, +0.506]
Top-p	1.5	2.330	**2.560**	2.540	2.500	[-0.130, +0.600]
Top-k	1.0	2.140	2.230	**2.380**	2.370	[-0.216, +0.406]
Top-k	1.5	2.740	2.930	2.680	**3.050**	[-0.186, +0.546]
Min-p	1.0	1.910	2.140	2.130	**2.150**	[-0.070, +0.530]
Min-p	1.5	2.490	2.630	**2.660**	2.520	[-0.200, +0.520]
P-less	1.0	1.080	**1.100**	1.070	1.090	[-0.060, +0.110]
P-less	1.5	1.250	1.290	**1.300**	**1.300**	[-0.100, +0.210]

D.4 Longer outputs: reasoning, instruction following, and code

Reasoning models, per model. MATH-500 uses two independent sets of eight answers per problem and AIME three, averaged within a problem. The pooled rows of Table 1 hide two per-model differences in opposite directions. On MATH-500, early release raises the pass@8 of DeepSeek-R1-Distill-Qwen 1.5B from 94.0% to 95.6% (+1.6 points, 95% interval [+0.6, +2.7]) and lowers that of the 7B model from 98.1% to 97.4% (-0.7 points, [-1.3, -0.1]), a difference of about three problems at a ceiling of 98%. Qwen3 1.7B is unchanged (+0.0 points, [-0.7, +0.8]). Exact stratification cannot lower expected pass@K, and we find no mechanism for the 7B decrease. Among the many intervals of the long-output settings, one excluding zero in each direction is within what chance produces. All AIME intervals include zero.

Table 12: **Every setting of Table 1.** Pass@1 and pass@K (%) per model, benchmark, truncation rule, and number of answers. R marks reasoning models. Code settings are in Table 13.

				Independent		Early release		Lattice	
Benchmark	K	Model	Rule	Pass@1	Pass@K	Pass@1	Pass@K	Pass@1	Pass@K
MATH-500	2	Qwen2.5 7B	Top-p	76.8	83.8	75.4	82.0	77.0	84.0
			Top-k	73.4	81.0	73.9	80.6	73.8	81.2
			Min-p	74.5	82.4	75.5	83.0	76.8	83.8
			P-less	76.5	80.8	77.3	83.0	77.7	82.8
MATH-500	2	Qwen2.5-Math 1.5B	Top-p	74.2	80.4	74.1	81.0	74.7	81.8
			Top-k	73.7	81.2	72.4	79.2	75.2	80.6
			Min-p	74.6	81.6	73.9	81.4	74.1	79.8
			P-less	75.8	79.0	75.6	79.8	76.3	80.2
MATH-500	4	Qwen2.5 7B	Top-p	76.0	87.4	76.5	86.8	75.4	87.4
			Top-k	73.2	86.0	74.5	86.0	73.7	87.6
			Min-p	75.3	88.4	75.7	86.4	75.8	87.6
			P-less	76.8	83.8	77.7	86.4	77.1	85.8
MATH-500	4	Qwen2.5-Math 1.5B	Top-p	75.0	86.0	74.4	86.8	75.5	85.0
			Top-k	73.1	85.8	73.5	86.4	73.0	86.0
			Min-p	74.8	86.4	74.7	86.0	74.0	84.8
			P-less	75.3	81.2	75.0	82.0	75.5	82.8
MATH-500	8	Qwen2.5 7B	Top-p	75.8	91.0	75.5	91.2	75.2	90.2
			Top-k	74.1	89.8	74.2	90.2	74.0	90.2
			Min-p	75.2	91.0	74.7	90.8	75.9	90.4
			P-less	76.7	86.2	76.8	87.0	77.0	87.0
MATH-500	8	Qwen2.5-Math 1.5B	Top-p	74.8	90.0	74.8	90.2	74.0	88.0
			Top-k	73.5	91.0	73.7	89.2	73.1	89.2
			Min-p	74.2	90.4	74.2	89.4	74.6	89.0
			P-less	75.9	84.8	75.6	84.0	74.9	83.2
GSM8K	8	Qwen2.5 7B	Top-p	91.3	97.0	91.7	97.3	91.3	97.1
			Top-k	91.0	97.1	90.8	97.0	90.9	97.5
			Min-p	91.0	97.1	91.3	97.0	91.1	97.3
			P-less	92.2	95.1	92.3	94.9	92.4	95.5
GSM8K	8	Qwen2.5-Math 1.5B	Top-p	84.6	95.6	84.9	95.4	85.2	95.6
			Top-k	84.9	95.9	84.6	95.6	84.9	95.5
			Min-p	85.1	95.1	85.1	94.9	85.2	95.1
			P-less	85.1	91.2	85.6	91.6	85.3	91.7
IFEval	8	Qwen2.5 0.5B	Top-p	25.7	51.6	25.5	51.6	25.6	52.9
			Top-k	25.4	51.2	25.7	52.9	26.1	52.7
			Min-p	26.8	52.9	26.7	51.9	26.7	54.5
			P-less	26.8	47.3	27.1	47.1	27.1	47.7
IFEval	8	Qwen2.5 7B	Top-p	70.8	84.5	70.4	84.5	70.6	85.6
			Top-k	70.4	85.8	70.4	85.8	70.1	84.5
			Min-p	71.5	84.7	71.1	84.3	71.3	85.0
			P-less	72.2	83.0	71.3	81.5	72.3	83.4
MATH-500, R	8	Qwen3 1.7B (thinking)	Top-p 0.95	88.1	95.7	88.5	95.7	–	–
MATH-500, R	8	DeepSeek-R1-Distill-Qwen 1.5B	Top-p 0.95	82.8	94.0	83.0	95.6	–	–
MATH-500, R	8	DeepSeek-R1-Distill-Qwen 7B	Top-p 0.95	92.5	98.1	92.4	97.4	–	–
AIME 24–25, R	8	Qwen3 1.7B (thinking)	Top-p 0.95	42.4	66.1	42.3	68.3	–	–
AIME 24–25, R	8	DeepSeek-R1-Distill-Qwen 1.5B	Top-p 0.95	25.9	48.9	26.0	51.1	–	–
AIME 24–25, R	8	DeepSeek-R1-Distill-Qwen 7B	Top-p 0.95	47.1	71.7	47.8	71.1	–	–

E Inside the Model: Where Early Release Acts

Early release changes only which tokens are sampled, and only for the first few tokens of an answer. We ask where in the model's computation those tokens are decided, and whether the more diverse answer sets also differ in the model's internal representations. We use two standard tools: a logit lens over the residual stream (Belrose et al., 2025) and the public BatchTopK sparse autoencoders (SAEs)

Table 13: **Code, every setting.** Pass@1 and pass@8 (%) on all HumanEval+ and MBPP+ problems with the extended EvalPlus tests, eight answers, temperature 1, averaged over two independent sets per problem.

		HumanEval+				MBPP+			
		Independent		Early release		Independent		Early release	
Model	Rule	P@1	P@8	P@1	P@8	P@1	P@8	P@1	P@8
SmolLM2 360M	Top-p	11.9	26.5	11.1	22.9	19.6	45.9	20.4	47.0
	Top-k	8.2	20.1	8.3	19.8	14.4	40.7	15.2	40.7
	Min-p	11.1	25.9	11.7	25.6	19.1	44.8	19.9	46.8
	P-less	15.4	19.2	15.7	20.7	27.6	35.7	27.5	36.2
Qwen2.5 0.5B	Top-p	24.6	49.4	25.1	51.2	32.9	60.3	32.6	59.5
	Top-k	20.7	44.8	20.5	43.9	26.5	56.1	27.1	58.2
	Min-p	24.5	48.5	24.7	51.5	31.5	58.1	32.6	60.8
	P-less	30.9	42.7	30.9	44.2	40.2	51.9	40.1	51.5
Llama 3.2 1B	Top-p	27.9	54.3	26.9	53.7	31.8	61.0	31.2	59.9
	Top-k	22.6	48.8	23.4	50.9	26.7	57.4	27.1	58.3
	Min-p	27.5	57.0	27.1	54.0	31.5	61.1	30.8	59.7
	P-less	34.6	44.8	34.9	45.7	38.1	45.2	37.9	46.3
Qwen2.5 1.5B	Top-p	44.0	75.9	44.2	75.0	49.9	76.1	49.5	76.5
	Top-k	35.4	72.9	34.4	69.5	43.0	73.9	43.8	73.1
	Min-p	43.4	73.8	43.6	75.0	49.4	75.3	49.3	75.1
	P-less	50.5	61.9	51.3	63.1	56.0	66.4	56.2	66.7
Qwen2.5-Coder 1.5B	Top-p	57.7	84.1	59.0	84.1	53.8	76.7	54.2	75.7
	Top-k	52.6	83.8	52.2	82.9	48.5	75.3	49.4	76.7
	Min-p	57.4	84.1	59.0	85.1	53.2	76.1	53.7	76.3
	P-less	65.0	73.5	64.8	73.5	60.1	67.9	60.0	68.4
SmolLM2 1.7B	Top-p	24.5	50.6	25.7	51.5	36.8	64.8	37.9	63.4
	Top-k	20.0	46.6	20.2	47.6	30.3	61.6	31.7	63.1
	Min-p	24.1	51.8	23.7	51.5	35.8	64.6	37.3	66.1
	P-less	29.8	38.7	29.7	37.8	45.5	57.7	45.3	58.3
Qwen2.5 7B	Top-p	75.2	89.0	74.7	88.4	68.6	80.0	67.6	80.6
	Top-k	74.2	89.9	74.4	89.0	67.7	81.1	67.9	81.1
	Min-p	75.1	90.2	75.1	89.6	68.1	80.0	67.9	80.3
	P-less	74.5	77.7	75.0	78.4	68.3	72.0	68.6	72.0
Qwen2.5-Coder 7B	Top-p	79.1	91.2	79.5	91.8	65.7	84.8	65.3	84.1
	Top-k	75.8	91.8	76.5	91.8	61.4	84.5	61.2	84.5
	Min-p	78.5	91.2	79.2	90.2	65.4	84.5	65.3	84.3
	P-less	83.0	86.9	82.9	86.6	70.1	75.3	70.0	76.1

for Qwen2.5 7B Instruct at layers 3, 7, 11, 15, 19, 23, and 27 (Cunningham et al., 2024, Bussmann et al., 2024). We generate eight answers per set with independent sampling and with early release for all 145 CoverageQA questions (three seeds) and all 100 curated NoveltyBench requests, at min-p 0.05 and temperature 1, and feed each answer back through the model to read the residual stream after every block over its first 32 tokens. A feature is active on a token when its encoder output exceeds the SAE's inference threshold.

Release happens at the model's forks. The logit lens applies the final normalization and unembedding to each block's output. The token whose choice places an answer's interval inside its slice is one where the model leaves the choice open: the median entropy of its final next-token distribution is 1.6 bits on NoveltyBench, against 0.1 bits at the nearly deterministic coordinated tokens before it. On CoverageQA, whose answers begin with the entity itself, the contrast is smaller (3.2 against 2.6 bits). The generated token gains its probability only in the last blocks (Figure 14a). Coordination therefore acts on the few open decisions that separate one answer from another and stops right after them.

Coordinated answers differ inside the model. For each set we compare the answers' residual streams, averaged over their first 32 tokens, by mean pairwise cosine distance, and their SAE feature sets, the features active on any of those tokens, by mean pairwise Jaccard overlap and by the number

of distinct features in the set. Early-release answers are significantly further apart at every block on CoverageQA and at 20 of 28 blocks on NoveltyBench (Figure 14b), and share about 4% fewer SAE features at every SAE layer on both benchmarks, with all 14 intervals below zero (Figure 14c). On CoverageQA an early-release set activates 6% more distinct features at layer 19 (95% interval [+3, +10]%), with intervals above zero at all seven layers. The extra diversity thus reaches the model's internal features, beyond the wording of the answers.

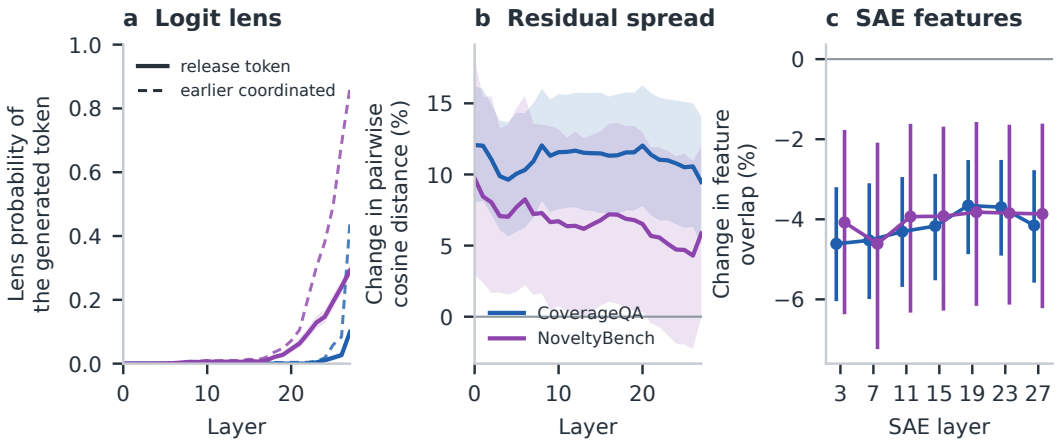

Figure 14: **Where early release acts inside Qwen2.5 7B Instruct.** (a) Logit lens: the probability each block's output assigns to the token that was generated, at the token that triggers release (solid) and at the coordinated tokens before it (dashed), averaged over answers. (b) Change in the mean pairwise cosine distance between the eight answers' residual streams (averaged over each answer's first 32 tokens), early release minus independent sampling, relative to independent sampling. (c) Change in the mean pairwise Jaccard overlap between the answers' SAE feature sets. Bands and bars are 95% paired prompt-bootstrap intervals. CoverageQA uses 145 questions with three seeds and NoveltyBench 100 requests.