NASH EQUILIBRIUM TEXT: A GAME-THEORETIC DECODING FRAMEWORK FOR TEXT GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Text generation is conventionally defined by a decoding procedure: autoregressive models commit to tokens from left to right, whereas masked and diffusion models construct sequences iteratively. These procedures do not necessarily generate sequences in which each token has high conditional probability given the prompt and the completed context. We formulate text generation as a finite game in which token positions are players, vocabulary items are actions, and each player's utility is the language model's conditional probability. A sequence is a Nash equilibrium when no position can increase its conditional probability through a unilateral token replacement. We show that Nash equilibria can have exponentially higher sequence probability than autoregressive outputs as sequence length grows. We further propose Nash decoding, which monotonically increases sequence probability by repeatedly replacing the most unstable token, and prove that it reaches an ε-Nash equilibrium in $O(\log(1/p(x^{(0)} \mid \text{prompt}))/\varepsilon)$ updates, where $x^{(0)}$ is the initial sequence. Empirically, Nash decoding consistently improves generations from masked language models, yielding higher likelihood and stronger downstream performance without additional training or fine-tuning. Across the CLAPNQ, PubMedQA, and CoQA datasets, Nash decoding with BERT-family models outperforms autoregressive models up to $18\times$ larger on F1 and ROUGE metrics, at the cost of additional test-time computation.

1 INTRODUCTION

"The first draft of anything is shit," Ernest Hemingway reportedly remarked (Samuelson, 1984). The first draft of this manuscript was no exception. Accomplished writers, at least before the advent of LLMs, have devoted considerable effort to revision. Hemingway, for instance, rewrote the final page of *A Farewell to Arms* 39 times (Plimpton, 1958). This raises a scientific question:

When does the revision terminate, and what is the equilibrium text that editing converges to?

Revision has become the core in text generation with language models where the first draft often is the standard de facto autoregressive text obeying

$$x_{i+1}^{(\mathrm{AR})} = \arg\max_{x \in \mathcal{V}} p(x \mid x_{\leq i}^{(\mathrm{AR})}; \text{prompt}), \tag{1}$$

where p denotes the probability function (Radford et al., 2019) and x_i are tokens selected from a vocabulary $\mathcal{V}$. Autoregressive generation calls for revision because each token is produced without access to the tokens that follow. Existing approaches to revision fall into four broad families: (1) *remasking* tokens in diffusion and masked language models (Ghazvininejad et al., 2019; Sahoo et al., 2024; Shi et al., 2024); (2) *reinforcement learning* with external feedback (Ouyang et al., 2022; OpenAI, 2024; Guo et al., 2025); (3) *Markov chain Monte Carlo (MCMC)* sampling (Karan & Du, 2026; Zhao et al., 2026; Xu et al., 2025); and (4) *hybrid* decoding techniques (Xia et al., 2017; Lee et al., 2018; Gu et al., 2019; Welleck et al., 2023). We focus on revision driven solely by conditional probabilities without external feedback. Such revision need not reach an equilibrium; the text may continue to change indefinitely. We therefore formulate revision as a game that admits an equilibrium.

The game: Given a prompt, n players are the positions $i \in [n]$, and each player's action is a token from $\mathcal{V}$ and player i receives utility $\log p(x_i \mid x_{-i}; \text{prompt})$, where x_{-i} denotes all tokens except x_i.

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.

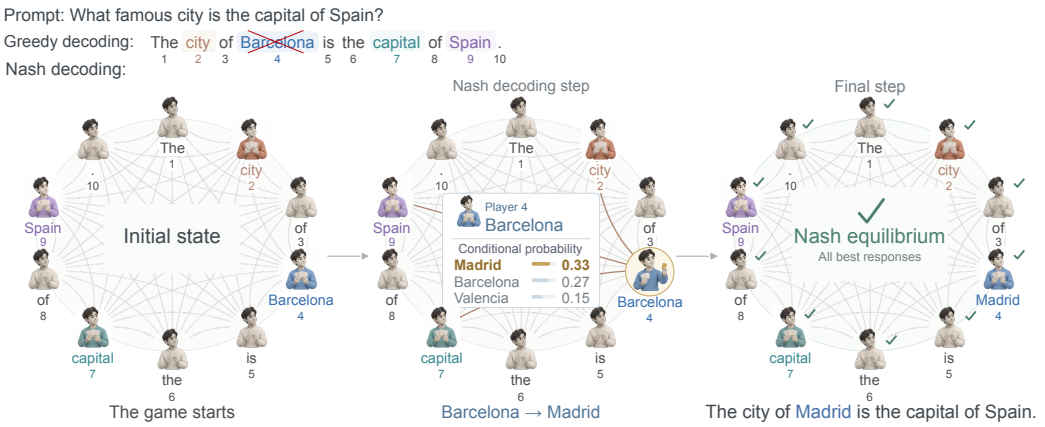


Figure 1: **Text generation as a game between token positions.** Each token position is a player whose choice is evaluated given all context tokens. Autoregressive decoding leaves earlier tokens fixed, whereas Nash decoding can use the same backbone to revise them using the completed context. We used GPT-2 small for conditional probability estimates. Each revision changes the context for the other players. At a Nash equilibrium, no player can increase its full-context conditional probability by changing its token. Details are reported in Appendix A.

Since $p(x_i \mid x_{-i}; \text{prompt}) = p(x_1, \ldots, x_n \mid \text{prompt}) / p(x_{-i} \mid \text{prompt})$ and the denominator does not depend on x_i, this is an exact potential game with potential $\log p(x_1, \ldots, x_n \mid \text{prompt})$ (Monderer & Shapley, 1996). Being finite, the game admits at least one pure Nash equilibrium $x^{(\text{nash})} = (x_1^{(\text{nash})}, \ldots, x_n^{(\text{nash})})$, which satisfies

$$\forall i \in [n]: \quad x_i^{(\text{nash})} = \arg\max_{x \in \mathcal{V}} \; p\big(x \mid x_{-i}^{(\text{nash})}; \text{prompt}\big). \tag{2}$$

In other words, a sequence is a *Nash equilibrium* when no single position can increase its conditional probability by changing its token. Comparing to Equation 1, the above conditioning is on x_{-i} rather than $x_{<i}$.

As an example, we simulate this revision game between tokens using GPT-2 prompted by the question "What famous city is the capital of Spain?". Figure 1 shows revisions of the first draft generated by autoregressive decoding that is "The city of Barcelona is the capital of Spain." Applying best-response updates replaces "Barcelona" with "Madrid", increasing the sequence conditional probability by a factor of 1.26[1]. We will show that Nash equilibria can attain substantially higher likelihood than autoregressive generations: the ratio $\frac{p(x^{(\text{nash})}|\text{prompt})}{p(x^{(\text{AR})}|\text{prompt})}$ can grow exponentially in n.

Computing an exact pure Nash equilibrium in general potential games is PLS-complete (Fabrikant et al., 2004). In contrast, we propose a revision mechanism called Nash decoding that obtains an ϵ-Nash equilibrium in $O(1/\varepsilon)$ time. Furthermore, the conditional probability of revised text monotonically increases through the revision.

Empirically, we show that masked language models can be used to compute a Nash equilibrium. Thus, we evaluate the question-answering performance of Nash equilibria against autoregressive generation on question-answering benchmarks. At the cost of additional computation, iterative Nash updates allow a masked language model to surpass the F1 and ROUGE of autoregressive models with 18 times as many parameters on CoQA (Reddy et al., 2019), PubMedQA (Jin et al., 2019), and CLAPNQ (Rosenthal et al., 2025) datasets. We also report negative results showing that the statistical efficiency of equilibrium varies and depends on the initialization and model probability estimation error.

[1]In this example, the increase in probability happens to coincide with improved factual correctness; in general, higher probability does not ensure correctness.

2 TEXT GENERATION AS A GAME

Why a Nash equilibrium? We motivate finding a Nash equilibrium by proving that its likelihood can be significantly larger than that of autoregressive text.

Lemma 1 (Separation between greedy AR decoding and Nash equilibrium) *For every positive even integer n, there exists a strictly positive autoregressive distribution on $\{0,1\}^n$ such that*

$$\frac{p(x^{(\mathrm{nash})})}{p(x^{(\mathrm{AR})})} = \exp(\gamma n), \qquad \gamma := \frac{1}{2}\log\left(\frac{539}{289}\right) \approx 0.31164.$$

The proof is postponed to Appendix B. While the above result is proven from a specific joint probability function, we will later show that for every probability function there exists at least one Nash equilibrium with higher likelihood than the autoregressive output.

2.1 APPROXIMATE EQUILIBRIUM IS ATTAINABLE

We study when *revision terminates*. We prove that there exists a revision process that terminates at an approximate Nash equilibrium. To make this notion precise, we first define the *token gap* at position i as the largest increase in conditional probability that a single token replacement at that position can achieve:

$$g_i(x) = \max_{v \in \mathcal{V}} p(v \mid x_{-i}; \text{prompt}) - p(x_i \mid x_{-i}; \text{prompt}). \tag{3}$$

The *Nash gap* is the largest improvement available at any position: $G(x) = \max_{i\in[n]} g_i(x)$. The equilibrium condition for $x^{(nash)}$ in Equation 2 is therefore equivalent to $G(x^{(nash)}) = 0$: no position can improve its utility through a unilateral token replacement. More generally, for a tolerance $\varepsilon \geq 0$, x is an *additive ε-Nash equilibrium* if $G(x) \leq \varepsilon$: no single-token replacement improves its position's conditional probability by more than ε.

Recall the formulation of conditional density as

$$p(x_i \mid x_{-i}; \text{prompt}) = \frac{p(x \mid \text{prompt})}{p(x_{-i} \mid \text{prompt})}. \tag{4}$$

The denominator is positive and unchanged by a unilateral deviation. Thus, whenever a player increases its utility, the joint probability also increases. In game-theoretic terms, the joint probability is an *ordinal potential* (Monderer & Shapley, 1996). This property yields a bound on the number of improving token refinements.

We define *Nash decoding* as the maximum-gap best-response procedure in Algorithm 1. Given a tolerance $0 < \varepsilon \leq 1$, start from any completed sequence $x^{(0)} \in \mathcal{V}^n$. At iteration $k \geq 0$, compute $g_i(x^{(k)})$ for every position $i \in [n]$. If $G(x^{(k)}) \leq \varepsilon$, return $x^{(k)}$. Otherwise, select a position with the largest gap and replace its token with the best response leaving all other tokens unchanged. It terminates when the maximum Nash gap is at most ε, yielding an ε-Nash equilibrium.

Algorithm 1 Nash Decoding

Require: Prompt, initial sequence $x^{(0)} \in \mathcal{V}^n$, tolerance $0 < \varepsilon \leq 1$
Ensure: Sequence x satisfying $G(x) \leq \varepsilon$

$k \leftarrow 0$
while $G(x^{(k)}) > \varepsilon$ **do**
  Compute $g_i(x^{(k)})$ for all $i \in [n]$
  $G(x^{(k)}) \leftarrow \max_{i\in[n]} g_i(x^{(k)})$
  Select $i_k \in \arg\max_{i\in[n]} g_i(x^{(k)})$
  Set $x_{i_k}^{(k+1)} \in \arg\max_{v\in\mathcal{V}} p\Big(x_{i_k} = v \mid x_{-i_k}^{(k)}; \text{prompt}\Big)$
  Set $x_i^{(k+1)} \leftarrow x_i^{(k)}$ for all $i \neq i_k$
  $k \leftarrow k + 1$
end while
return $x^{(k)}$

The next theorem shows that Nash decoding monotonically increases the sequence probability at every performed update and reaches an ε-Nash equilibrium in a polynomial number of iterations.

Theorem 1 (Convergence Rate) *Let $0 < \varepsilon \leq 1$ and let $p_0 = p(x^{(0)} \mid \textit{prompt}) > 0$ be the initial sequence probability. Every performed update of the above dynamics satisfies*

$$\frac{p(x^{(k+1)} \mid \textit{prompt})}{p(x^{(k)} \mid \textit{prompt})} \geq 1 + \underbrace{G(x^{(k)})}_{>0}. \tag{5}$$

The dynamics terminate at an additive ε-Nash equilibrium after K_ε token replacements, with

$$K_\varepsilon \leq \frac{\log(1/p_0)}{\log(1+\varepsilon)} \leq \frac{2\log(1/p_0)}{\varepsilon}. \tag{6}$$

Proof. Let $p_k = p(x^{(k)} \mid \text{prompt})$. Equation 4 and maximum-gap selection imply

$$p_{k+1} - p_k = p(x^{(k)}_{-i_k} \mid \text{prompt}) G(x^{(k)}) \geq p_k G(x^{(k)}),$$

since the marginal probability is at least the probability of the completed sequence. This proves Equation 5. Each performed update has $G(x^{(k)}) > \varepsilon$, so any K performed updates satisfy

$$1 \geq p_K \geq p_0(1+\varepsilon)^K.$$

Taking logarithms bounds the number of updates; hence the dynamics must terminate with $G(x) \leq \varepsilon$. Finally, $\log(1+\varepsilon) \geq \varepsilon/2$ for $0 < \varepsilon \leq 1$. □

The bound depends logarithmically on the inverse initial probability: a more probable starting sequence yields a tighter guarantee. To distinguish replacements from their computational cost, count one *conditional query* as obtaining $p(\cdot \mid x_{-i}; \text{prompt})$ over the entire vocabulary at one position. Recomputing all gaps after each replacement requires $O\big(n \log(1/p_0)\varepsilon^{-1}\big)$ density evaluations, including the final scan that certifies termination.

A consequence of this result is: for every autoregressive output defined in Equation 1, there exists a Nash equilibrium whose likelihood is at least as high. This follows because Nash decoding initialized at the autoregressive text never decreases the likelihood.

2.2 Nash Decoding with Pretrained Language Models

Nash decoding relies on the conditional probability $p(v \mid x_{-i}; \text{prompt})$ of token v at position i given the rest of the sequence. This quantity can be estimated with either autoregressive or masked language models.

Masked language models. A masked language model (MLM), such as BERT, provides the conditional estimates needed for Nash decoding. For each position i, we replace x_i with `[MASK]` while retaining the prompt and all other tokens. The model produces a logit $z_{\theta,i}(v)$ for each candidate token $v \in \mathcal{V}$ at the masked position. Applying softmax gives

$$q_\theta(v \mid x_{-i}; \text{prompt}) = \frac{\exp(z_{\theta,i}(v))}{\sum_{w \in \mathcal{V}} \exp(z_{\theta,i}(w))},$$

where w ranges over the vocabulary and the logits depend on the masked input. Because the model is pretrained to reconstruct masked tokens from their surrounding context, we use this distribution as an estimate of $p(v \mid x_{-i}; \text{prompt})$ (Devlin et al., 2019). A single model evaluation provides both the highest-probability replacement and the estimated probability of the original token x_i, which together determine the token gap.

Autoregressive models. Nash decoding can also use an AR model, as illustrated by the motivating example in Section 1. An AR model defines a joint distribution $p^{\mathrm{AR}}_\theta(x \mid \text{prompt})$ as the product of its left-to-right token probabilities, so its full-context conditional is

$$q^{\mathrm{AR}}_\theta(v \mid x_{-i}; \text{prompt}) = \frac{p^{\mathrm{AR}}_\theta(x_{<i}, v, x_{>i} \mid \text{prompt})}{\sum_{w \in \mathcal{V}} p^{\mathrm{AR}}_\theta(x_{<i}, w, x_{>i} \mid \text{prompt})}. \tag{7}$$

These conditionals are therefore compatible with the model's joint distribution by construction. However, computing them is expensive because each replacement can change subsequent token probabilities, requiring up to $|\mathcal{V}|$ candidate suffix evaluations per position.

The challenge of joint density estimation. The conditional distributions supplied by a masked language model need not be compatible with a common joint distribution (Torroba Hennigen & Kim, 2023) with a joint probability p where

$$\begin{aligned} p(x \mid \text{prompt}) &= p(x_i \mid x_{-i}; \text{prompt})\, p(x_{-i} \mid \text{prompt}) \\ &= p(x_j \mid x_{-j}; \text{prompt})\, p(x_{-j} \mid \text{prompt}) \end{aligned} \tag{8}$$

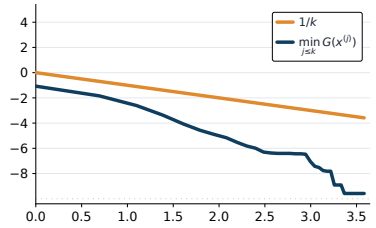
holds for every sequence x and every pair of positions i and j. Consequently, the potential argument and convergence guarantee in Theorem 1 do not automatically apply to these estimates. We write G_θ for the Nash gap computed from the estimated conditionals q_θ. A completed sequence with $G_\theta(x) = 0$ is stable under this estimated game, but this certificate does not establish global optimality or answer correctness. We therefore study convergence and generation quality empirically.

2.3 Empirical Behavior Predicted by Theorem 1

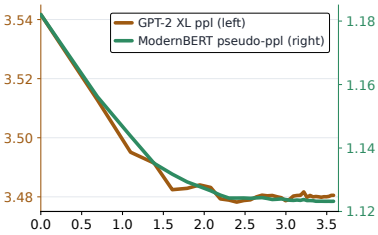
Masked-model conditionals may not satisfy the assumptions of our convergence theorem. We therefore examine whether updates converge in practice and how refinement affects likelihood diagnostics. Using ModernBERT (Warner et al., 2025), we generate 64-token continuations for 500 prompts from WikiText-103 (Merity et al., 2017). Each continuation starts from 64 `[MASK]` tokens and repeatedly replaces the maximum-gap token with its best response. The setup is described in Appendix C.

Equilibrium dynamics. Figure 2a compares the decay rate of the Nash gap with the established theoretical $1/k$ decay rate in Theorem 1, where k indexes updates. The empirical curve is plotted after the last `[MASK]` token is replaced and refinement of the completed sequence begins (Appendix C.4). The curve remains below the reference curve and follows a similar overall slope, consistent with the predicted decay pattern. The plateaus near the end are due to the fact that most texts have already reached equilibrium at a faster rate compared to the theoretical rate $O(1/k)$.

Likelihood diagnostics. Figure 2b reports ModernBERT pseudo-perplexity and continuation perplexity under the external autoregressive model GPT-2 XL (Radford et al., 2019). Both scores decrease overall during refinement and then stabilize. Although ModernBERT's conditional estimates need not arise from a common joint distribution, the observed decrease in pseudo-perplexity is consistent with the likelihood improvement established by Theorem 1.



(a) Equilibrium diagnostic.



(b) Likelihood diagnostics.

Figure 2: Empirical behavior of Nash decoding on WikiText-103. (a) Running minimum of the Nash gap, compared with a $1/k$ reference curve. The reference indicates a rate of decay, aligned with Theorem 1. Both axes are log-transformed. (b) GPT-2 XL continuation perplexity (left axis) and ModernBERT pseudo-perplexity (right axis). The x-axis is log-transformed.

3 Related Work

Nash decoding, introduced in this work, links to a family of decoding methods which we categorize into three main groups: (1) multi-pass decoders, (2) parallel decoders, and (3) speculative decoders.

(1) Multi-pass decoders. This work is inspired by a rich body of prior work on refining autoregressive text generation to improve the likelihood of the generated text. Xia et al. (2017) address this limitation with a first-pass autoregressive decoder that produces a complete draft and a trained second-pass decoder that refines it. The first generator might itself prefer to refine any token once the draft is complete. Self-Refine instead uses the same model to generate and revise its output (Madaan

et al., 2023), but its revisions target task-specific quality criteria rather than certifying token-level equilibrium. Zhou et al. (2026) define a competitive MCMC *sampling* (remasking) mechanism that bridges the gap between masked and autoregressive language models.

(2) Parallel decoders. Like masked diffusion language models (Sahoo et al., 2024; Shi et al., 2024), Nash decoding iteratively revises tokens. However, diffusion language models focus on parallel text generation (Austin et al., 2021; Nie et al., 2025), whereas our procedure for finding a Nash equilibrium is sequential, updating one token at a time. This reflects a difference in objectives: rather than computational efficiency of decoding, we focus on characterizing the likelihood properties of Nash-equilibrium sequences.

(3) Speculative decoders. Speculative decoding accelerates autoregressive generation by using a draft model to propose tokens and a target model to verify them (Leviathan et al., 2023; Chen et al., 2023). In SpecDec, a drafted token is revised if it falls outside the verifier's top tokens or if its log-likelihood is more than a threshold below that of the verifier's top prediction. At the first such position, the verifier replaces the token with its own prediction and discards the remaining draft (Xia et al., 2023). These methods verify tokens using prefix-conditioned predictions, after which accepted tokens remain fixed. In contrast, Nash decoding can revisit earlier positions and updates tokens using full-context best responses to seek a Nash equilibrium sequence.

A growing body of work studies game-theoretic interactions among language-model agents and response policies (Zhang et al., 2025; Jacob et al., 2024; Munos et al., 2024; Hagen et al., 2026), including test-time methods for human-value alignment (Zhang et al., 2024; Tamine et al., 2025). Nash learning from human feedback (Munos et al., 2024), for example, seeks an equilibrium between competing response policies. In contrast, we formulate decoding itself as a game in which positions within a single generated sequence are players and tokens are their actions.

4 Empirical Results and Benchmarking

We assess the question-answering performance of Nash decoding compared to existing autoregressive and masked diffusion decoding. For conditional probability estimates, we used four pretrained models: ModernBERT-Large (395M) (Warner et al., 2025), RoBERTa-Large (355M) (Liu et al., 2019), mmBERT-base (Marone et al., 2026), and Ettin-400M (396M) (Weller et al., 2025).

Datasets. We used three standard question-answering benchmarks akin to prior works (Reddy et al., 2019; Dong et al., 2019; Rosenthal et al., 2025; Luo et al., 2022; Huang et al., 2023):

(1) CLAPNQ (Rosenthal et al., 2025) pairs a Natural Questions passage with either a cohesive long-form answer or no answer at all; we use the 300 *answerable* questions from its development split (mean answer length 63.8 tokens).
(2) From PubMedQA (Jin et al., 2019), we use the 500 PubMedQA-Labeled test questions, where the model must infer the abstract's withheld conclusion rather than extract or repeat text from the input (mean answer length 48.3 tokens).
(3) From CoQA (Reddy et al., 2019), we use the 1,814 development questions whose reference answer contains at least 5 tokens (using ModernBERT tokenizer); this threshold excludes bare *yes*/*no* and other very short answers, leaving enough positions for meaningful token interactions (mean answer length 7.5 tokens) [2].

Appendix D describes how the long-answer questions were selected and reports their statistics.

Protocol. We use a fixed prompt format for all models within each dataset. For CLAPNQ and PubMedQA, the passage and question are followed by the soft instruction "This question is answered completely with evidence from the passage. Answer:" and an *Answer:* prefix. For CoQA, which consists of conversational question–answer turns, we instead use the standard *Q:* and *A:* dialogue format following the original evaluation setup (Radford et al., 2019). Across all datasets, every model receives the same oracle answer-length budget T, computed from the reference answer using that model's tokenizer. End-of-sequence and structural tokens are banned, so every model emits

[2] To verify that the ≥ 5-token filter does not drive the CoQA results, we also evaluate on the full development set, including all questions; see Appendix F.

exactly T tokens. For Nash decoding, the masked backbone is used without additional training: we append a canvas of T `[MASK]` tokens to the prompt and iteratively update tokens using the model's full-context conditionals until reaching equilibrium with $\varepsilon = 0$. We report SQuAD-style token F1 (Rajpurkar et al., 2016), scored against the most favorable reference, together with ROUGE-L and ROUGE-Lsum (Lin, 2004). Appendix E provides the complete protocol.

4.1 Question-Answering Performance

Table 1 reports the question-answering performance of masked language models with and without Nash decoding, alongside strong autoregressive baselines. With Nash decoding, masked language models become effective zero-shot question answerers, outperforming autoregressive models $18\times$ their size in F1, ROUGE-L, and ROUGE-Lsum. These gains come at the cost of additional test-time computation, which we analyze in Section 5.

Table 1: Results across three question-answering benchmarks under a shared evaluation protocol. **Size** denotes the number of model parameters; for Nash decoding, it denotes the masked backbone size. Best results are **bolded** and second-best results are underlined. The complete benchmark results are reported in Appendix F.

		CoQA			PubMedQA			CLAPNQ		
Model	**Size**	F1	R-L	R-Ls	F1	R-L	R-Ls	F1	R-L	R-Ls
One-shot masked LMs (not fine-tuned)										
RoBERTa-L (Liu et al., 2019)	355M	28.00	31.45	31.46	7.99	8.21	8.74	5.94	10.38	10.59
ModernBERT-L (Warner et al., 2025)	395M	39.54	42.92	42.93	5.93	7.95	8.34	7.03	12.05	12.66
Ettin-400m (Weller et al., 2025)	396M	43.87	46.22	46.24	4.03	4.55	4.76	6.97	11.78	12.41
Autoregressive models										
OPT-350M (Zhang et al., 2022)	331M	27.16	29.86	29.94	23.43	20.43	21.41	33.21	32.14	33.39
SmolLM2-360M (Allal et al., 2025)	362M	52.77	54.30	54.32	21.56	19.71	20.57	32.62	31.02	32.66
GPT-2 L (Radford et al., 2019)	774M	42.61	44.31	44.38	22.09	19.85	20.84	33.06	31.50	33.68
GPT-2 XL (Radford et al., 2019)	1.5B	46.83	48.26	48.35	23.78	20.77	22.00	30.58	29.51	30.73
BLOOM (BigScience et al., 2022)	3B	56.69	56.25	56.62	22.60	19.53	21.46	34.00	32.18	34.31
OLMo-1.7 (Groeneveld et al., 2024)	7B	51.44	53.53	53.57	19.67	17.71	20.09	30.12	28.63	32.15
Nash decoding with same masked LMs										
RoBERTa-L	355M	41.35	44.02	44.04	**24.99**	**21.17**	**22.69**	25.25	25.59	25.17
ModernBERT-L	395M	58.12	58.71	58.71	24.28	20.33	21.36	**38.04**	**34.79**	**37.42**
Ettin-400m	396M	**61.79**	**61.98**	**62.03**	22.77	20.05	21.65	34.04	31.61	33.84

While Nash decoding turns frozen masked language models into zero-shot text generators without additional training or fine-tuning, the model performance varies across datasets, showing that Nash decoding remains dependent on the underlying model used for conditional density estimation. Better estimation leads to better performance.

Remarkably, the results in Table 1 must not be compared directly with dataset leaderboards, which typically include models fine-tuned on task-specific training data. For example, fine-tuned models on CoQA reach 90.7 F1 (Ju et al., 2019). In contrast, every model in Table 1 is evaluated zero-shot. Our comparison therefore isolates decoding methods with pretrained models under a shared evaluation protocol rather than targeting state-of-the-art performance.

For reference, Radford et al. (2019) report that GPT-2 XL achieves approximately 54.8 F1 on all CoQA questions. Our lower score reflects the subset used in Table 1, which contains only questions with answers of at least five tokens and thus excludes yes/no and other short answers. We exclude these because, for a single-token answer, the Nash equilibrium coincides with autoregressive decoding, leaving no room for revision.

4.2 Ablation Analysis

We study how token replacements evolve as the sequence approaches equilibrium, whether local equilibrium conditions preserve sequence-level coherence, and how the initialization determines which equilibrium is ultimately reached. Additional ablations are provided in Appendix G.

Effect of Token Replacements. Figure 3 tracks F1 score and the Nash gap throughout Nash decoding with ModernBERT-Large on CLAPNQ. F1 measures movement toward the reference answer, whereas the Nash gap measures the remaining incentive for any player to change its token. Generation starts from an entirely masked sequence, whereas ModernBERT was pretrained with a 30% masking rate (Warner et al., 2025). With little initial context, the model is uncertain, but as early tokens are filled, the emerging context sharpens the full-context conditionals and the maximum gap can increase. Once sufficient context is placed, Nash updates increasingly resolve unstable positions and drive the Nash gap toward zero. F1 continues to increase throughout the trajectory, but more slowly in the second half as most examples reach equilibrium, leaving only a smaller subset with profitable token deviations (more information is reported in Appendix G.1).

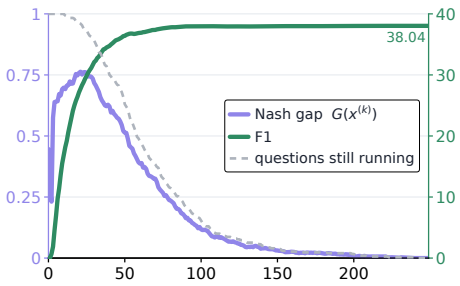


Figure 3: F1 score and Nash gap during Nash decoding iterations on CLAPNQ.

Qualitative example of Nash equilibrium. Beyond aggregate scores, we inspect a representative Nash-equilibrium answer on CLAPNQ. For the question *"where do they shoot guy's grocery games"*, Nash decoding using ModernBERT-Large fills a 48-slot canvas and returns,

> *"Guy's Grocery Games Season 1 was shot inside of an actual grocery store, Field's Market in West Hills, California. For Season 2, the market was built in a 15,500 square foot warehouse in Santa Rosa."*

It contains two well-formed sentences with proper nouns and numbers, scoring 98.5 F1, showing that Nash equilibrium can be a coherent grounded text without *model fine-tuning* (Appendix G.2).

Initialization and Equilibrium Basins. Our earlier results show that Nash-equilibrium outputs can be high-quality answers. Here, we show that the token game can admit multiple equilibria, and that the equilibrium reached depends on both the conditional estimator and the initialization. Table 2 shows that equilibrium quality depends on the quality of the conditional estimator: more accurate full-context conditionals $q_\theta(v \mid x_{-i}; \text{prompt})$ can define higher-quality stable sequences, whereas a zero Nash gap only certifies stability under the model's estimated token game. Initializations can lead to different equilibria because the token game can contain multiple basins of attraction. We also compare two constructions in Table 2. In the *left-to-right* construction, tokens are committed sequentially from left to right, with each generated token used as context for predicting the next; Nash decoding then refines the completed sequence until reaching an equilibrium. In *all-mask*, generation starts from a fully masked answer, and Nash decoding refines the sequence directly from this state. Therefore, the conditional estimator determines the equilibrium landscape, while the initialization determines the trajectory through that landscape.

Table 2: Nash Equilibria Basins on PubMedQA.

Model	Size	F1	R-L	R-Ls
Left-to-right initialization				
mmBERT-base	308M	4.22	6.11	5.38
RoBERTa-Large	355M	22.23	19.79	20.77
ModernBERT-Large	395M	21.91	19.88	19.99
Ettin-400m	396M	17.41	17.29	17.36
All-mask construction				
mmBERT-base	308M	3.89	5.30	4.59
RoBERTa-Large	355M	**24.99**	**21.17**	**22.69**
ModernBERT-Large	395M	24.28	20.33	21.36
Ettin-400m	396M	22.77	20.05	21.65

4.3 From Diffusion Updates to Equilibrium Updates

Masked diffusion models generate text through iterative full-context refinement, typically updating multiple masked positions in parallel. The closest existing method to Nash decoding is the particularly strong single-token masked diffusion model studied in dLLM (Zhou et al., 2026). dLLM provides a powerful instruction-tuned masked model, `ModernBERT-chat`, together with iterative decoding procedures that construct text through full-context token predictions. We analyze whether the same instruction-tuned model can benefit further when token updates are driven to a Nash equilibrium. Both approaches start from an all-`[MASK]` sequence and update one position at a time; dLLM selects masked positions using uniform or confidence-based ordering, whereas

Nash decoding selects the position with the largest Nash gap and can revisit previously generated tokens. Because both methods can use the same masked language model, this provides a controlled comparison of the position-selection and revision rules.

Table 3: Nash decoding versus single-token masked-diffusion decoding using the same masked model. All rows use the instruction-tuned `ModernBERT-chat` with its corresponding prompt; details in Appendix H.

	CoQA			PubMedQA			CLAPNQ		
Decoding rule	F1	R-L	R-Ls	F1	R-L	R-Ls	F1	R-L	R-Ls
Uniform-ordered	38.65	40.05	40.08	30.50	23.51	27.14	56.15	50.08	55.00
Confidence-ordered	41.85	42.21	42.28	31.74	24.64	28.50	58.84	54.30	58.11
Max Nash gap	**43.07**	**43.22**	**43.31**	**32.11**	**24.80**	**28.75**	**59.16**	**54.67**	**58.37**

Table 3 compares Nash decoding with the two decoding rules used by dLLM while holding the model, initialization, prompt, and generation budget fixed. All methods use the released `ModernBERT-chat` checkpoint, which is based on ModernBERT-Large and supervised fine-tuned by dLLM for masked generation. dLLM's *uniform* rule selects a masked position at random, whereas its *confidence* rule commits the position whose predicted token has the highest conditional probability; committed tokens are not revisited. Nash decoding instead continues revising tokens until reaching equilibrium. Across all three datasets, equilibrium-based decoding with the dLLM fine-tuned model yields higher F1 and ROUGE scores, showing that these strong diffusion-trained models can support even higher-quality generations when their outputs are driven to token-level equilibrium, though at the cost of additional model calls and test-time computation.

5 Discussion and Limitations

Our paper establishes the potential of game-theoretic decoding by proving that it runs in polynomial time and by demonstrating its performance on question-answering benchmarks. These results call for future work that addresses the limitations of Nash decoding, both theoretically and practically.

On the theoretical side, the convergence of Nash decoding established in Theorem 1 requires full-context conditionals that are compatible with a common joint distribution. However, the conditional distributions estimated by masked language models do not necessarily satisfy this requirement, since training does not enforce compatibility conditions such as those in Equation 8. A promising direction for future research is to enforce these conditions, which would provide a provable convergence guarantee to an approximate Nash equilibrium.

A second limitation is computational cost. Although Nash decoding may require only a modest number of token replacements to reach equilibrium, each replacement is expensive with current masked architectures because computing the maximum Nash gap requires evaluating leave-one-out conditionals across all n token positions. Consequently, the number of model calls per refinement step grows linearly with the generation length n, and the total query cost scales as $O(nK)$ for K token replacements. Figure 4 illustrates this overhead on CoQA, our shortest-answer benchmark. The statistical gains of Nash decoding currently come at substantially higher test-time computation, and the gap becomes more pronounced for longer generations. This overhead is primarily architectural rather than inherent to the equilibrium objective: masked models that can evaluate full-context conditionals for multiple positions more efficiently could substantially reduce the cost.

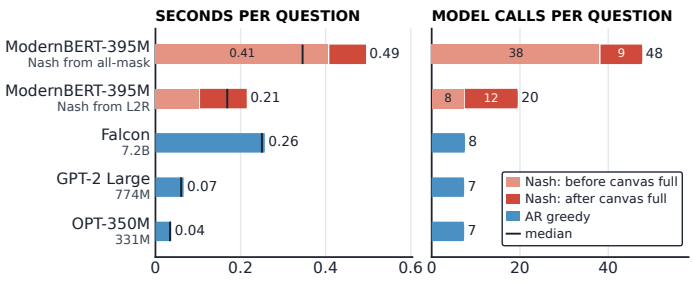


Figure 4: Cost of Nash decoding

REFERENCES

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 small language model, 2025. URL https://arxiv.org/abs/2502.02737.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. The falcon series of open language models, 2023. URL https://arxiv.org/abs/2311.16867.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/958c530554f78bcd8e97125b70e6973d-Abstract.html.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 2397–2430, 2023.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. BLOOM: A 176B-parameter open-access multilingual language model, 2022. URL https://arxiv.org/abs/2211.05100.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling, 2023. URL https://arxiv.org/abs/2302.01318.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, volume 32, pp. 13063–13075, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/c20bb2d9a50d5ac1f713f8b34d9aac5a-Abstract.html.

Alex Fabrikant, Christos H. Papadimitriou, and Kunal Talwar. The complexity of pure nash equilibria. In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*, pp. 604–612. ACM, 2004. doi: 10.1145/1007352.1007445.

Xinyang Geng and Hao Liu. OpenLLaMA: An open reproduction of LLaMA. GitHub repository, May 2023. URL https://github.com/openlm-research/open_llama.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-Predict: Parallel decoding of conditional masked language models. 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. 6112–6121. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1633. URL https://aclanthology.org/D19-1633/.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, William Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah Smith, and Hannaneh Hajishirzi. OLMo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15789–15809, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.841. URL https://aclanthology.org/2024.acl-long.841/.

Jiatao Gu, Changhan Wang, and Junbo Zhao. Levenshtein Transformer. In *Advances in Neural Information Processing Systems*, volume 32, pp. 11179–11189. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/675f9820626f5bc0afb47b57890b466e-Abstract.html.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Luca Hagen, Johanna P Müller, Weitong Zhang, Mengyun Qiao, and Bernhard Kainz. Wasserstein equilibrium decoding for reliable medical visual question answering. *arXiv preprint arXiv:2605.18313*, 2026.

Shaoyao Huang, Luozheng Qin, and Ziqiang Cao. Diffusion language model with query-document relevance for query-focused summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 11020–11030, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.735. URL https://aclanthology.org/2023.findings-emnlp.735/.

IBM Research. Granite-7B-Base. Hugging Face model card, 2024. URL https://huggingface.co/ibm-granite/granite-7b-base.

Athul Paul Jacob, Yikang Shen, Gabriele Farina, and Jacob Andreas. The consensus game: Language model generation via equilibrium search. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=n9xeGcI4Yg.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. PubMedQA: A dataset for biomedical research question answering. 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. 2567–2577. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1259. URL https://aclanthology.org/D19-1259/.

Ying Ju, Fubang Zhao, Shijie Chen, Bowen Zheng, Xuefeng Yang, and Yunfeng Liu. Technical report on conversational question answering, 2019. URL https://arxiv.org/abs/1909.10772.

Aayush Karan and Yilun Du. Reasoning with sampling: Your base model is smarter than you think. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/01498f17142547dadb60999c39bd0a09-Abstract-Conference.html.

Wooseok Kim, Gyunyeop Kim, and Sangwoo Kang. Accelerating inference in retrieval-augmented generation models for long-form question answering via dynamic token pruning. *Mathematics*, 13(14):2231, 2025. doi: 10.3390/math13142231. URL https://www.mdpi.com/2227-7390/13/14/2231.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1173–1182, 2018. doi: 10.18653/v1/D18-1149. URL https://aclanthology.org/D18-1149/.

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 of *Proceedings of Machine Learning Research*, pp. 19274–19286. PMLR, 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Zhengzhong Liu, Aurick Qiao, Willie Neiswanger, Hongyi Wang, Bowen Tan, Tianhua Tao, Junbo Li, Yuqi Wang, Suqi Sun, Omkar Pangarkar, Richard Fan, Yi Gu, Victor Miller, Yonghao Zhuang, Guowei He, Haonan Li, Fajri Koto, Liping Tang, Nikhil Ranjan, Zhiqiang Shen, Xuguang Ren, Roberto Iriondo, Cun Mu, Zhiting Hu, Mark Schulze, Preslav Nakov, Tim Baldwin, and Eric P. Xing. LLM360: Towards fully transparent open-source LLMs. 2023.

Llama Team, AI @ Meta. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. BioGPT: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6):bbac409, 2022. doi: 10.1093/bib/bbac409. URL https://academic.oup.com/bib/article/23/6/bbac409/6713511.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-2019. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91edff07232fb1b55a505a9e9f6c0ff3-Abstract-Conference.html.

Marc Marone, Orion Weller, William Fleshman, Eugene Yang, Dawn Lawrie, and Benjamin Van Durme. mmBERT: A modern multilingual encoder with annealed language learning. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2509.06888.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *5th International Conference on Learning Representations (ICLR 2017), Conference Track Proceedings*, Toulon, France, 2017. OpenReview.net. URL https://openreview.net/forum?id=Byj72udxe.

Dov Monderer and Lloyd S Shapley. Potential games. *Games and economic behavior*, 14(1):124–143, 1996.

Remi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Côme Fiegel, Andrea Michi, Marco Selvi, Sertan Girgin, Nikola Momchev, Olivier Bachem, Daniel J. Mankowitz, Doina Precup, and Bilal Piot. Nash learning from human feedback. In *Proceedings of the 41st*

International Conference on Machine Learning, volume 235 of *Proceedings of Machine Learning Research*, pp. 36743–36768. PMLR, 2024. URL https://proceedings.mlr.press/v235/munos24a.html.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025. doi: 10.48550/arXiv.2502.09992. URL https://arxiv.org/abs/2502.09992.

OpenAI. Openai o1 system card, 2024. URL https://arxiv.org/abs/2412.16720.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Pascal Pfeiffer, Philipp Singer, Yauhen Babakhin, Gabor Fodor, Nischay Dhankhar, and Sri Satish Ambati. H2O-Danube3 technical report, 2024. URL https://arxiv.org/abs/2407.09276.

George Plimpton. Ernest hemingway, the art of fiction no. 21. *The Paris Review*, (18), 1958. URL https://www.theparisreview.org/interviews/4825/the-art-of-fiction-no-21-ernest-hemingway. Interview with Ernest Hemingway.

Qwen Team. Qwen2.5 technical report, 2024. URL https://arxiv.org/abs/2412.15115.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, 2016. URL https://aclanthology.org/D16-1264/.

Siva Reddy, Danqi Chen, and Christopher D. Manning. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019. doi: 10.1162/tacl_a_00266. URL https://aclanthology.org/Q19-1016/.

Sara Rosenthal, Avirup Sil, Radu Florian, and Salim Roukos. CLAPnq: Cohesive long-form answers from passages in natural questions for RAG systems. *Transactions of the Association for Computational Linguistics*, 13:53–72, 2025. doi: 10.1162/tacl_a_00729. URL https://aclanthology.org/2025.tacl-1.3/.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 130136–130184, 2024. doi: 10.52202/079017-4135. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/eb0b13cc515724ab8015bc978fdde0ad-Abstract-Conference.html.

Arnold Samuelson. *With Hemingway: A Year in Key West and Cuba*. Random House, New York, 1984. ISBN 9780394539836.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3277. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/bad233b9849f019aead5e5cc60cef70f-Abstract-Conference.html.

Mélissa Tamine, Otmane Sakhi, and Benjamin Heymann. Data valuation for llm fine-tuning: Efficient shapley value approximation via language model arithmetic. *arXiv preprint arXiv:2512.15765*, 2025.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Michal Guerquin, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2 OLMo 2 Furious, 2024. URL https://arxiv.org/abs/2501.00656.

Lucas Torroba Hennigen and Yoon Kim. Deriving language models from masked language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1149–1159, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.99. URL https://aclanthology.org/2023.acl-short.99/.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2526–2547, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.127. URL https://aclanthology.org/2025.acl-long.127/.

Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. Generating sequences by learning to self-correct. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=hH36JeQZDaO.

Orion Weller, Kathryn Ricci, Marc Marone, Antoine Chaffin, Dawn Lawrie, and Benjamin Van Durme. Seq vs Seq: An open suite of paired encoders and decoders, 2025. URL https://arxiv.org/abs/2507.11412.

Heming Xia, Tao Ge, Peiyi Wang, Si-Qing Chen, Furu Wei, and Zhifang Sui. Speculative decoding: Exploiting speculative execution for accelerating seq2seq generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 3909–3925, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.257. URL https://aclanthology.org/2023.findings-emnlp.257/.

Yingce Xia, Fei Tian, Lijun Wu, Jianxin Lin, Tao Qin, Nenghai Yu, and Tie-Yan Liu. Deliberation networks: Sequence generation beyond one-pass decoding. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/c6036a69be21cb660499b75718a3ef24-Abstract.html.

Dongling Xiao, Han Zhang, Yukun Li, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. ERNIE-GEN: An enhanced multi-flow pre-training and fine-tuning framework for natural language generation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pp. 3997–4003. International Joint Conferences on Artificial Intelligence Organization, 2020. doi: 10.24963/ijcai.2020/553. URL https://www.ijcai.org/proceedings/2020/553.

Minkai Xu, Tomas Geffner, Karsten Kreis, Weili Nie, Yilun Xu, Jure Leskovec, Stefano Ermon, and Arash Vahdat. Energy-based diffusion language models for text generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=sL2F9YCMXf.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: Open pre-trained transformer language models, 2022. URL https://arxiv.org/abs/2205.01068.

Weitong Zhang, Chengqi Zang, and Bernhard Kainz. Truth or deceit? a bayesian decoding game enhances consistency and reliability. *arXiv preprint arXiv:2410.01064*, 2024.

Weitong Zhang, Chengqi Zang, and Bernhard Kainz. Strategic llm decoding through bayesian games. In *ICLR 2025 Workshop on Reasoning and Planning for Large Language Models*, 2025.

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason without external rewards. In *International Conference on Learning Representations*, 2026.

Zhanhui Zhou, Lingjie Chen, Hanghang Tong, and Dawn Song. dLLM: Simple diffusion language modeling. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 78–88. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-demo.8. URL https://aclanthology.org/2026.acl-demo.8/.

TECHNICAL APPENDICES AND SUPPLEMENTARY MATERIAL

A THE MOTIVATING EXPERIMENT AND THE ILLUSTRATED TOKEN GAME

Setup. All numbers in this section are produced with GPT-2 Small (124M) in float32 with TF32 matrix shortcuts disabled, using greedy decoding. Generation and refinement use the *same* prompt. Conditionals are computed exactly over the full 50,257-token vocabulary, and refinement replaces, at each step, the single position with the largest gap, halting when the Nash gap falls below 10^{-9}.

First example: one revision. The example illustrated in the introduction concerns the prompt

Prompt

What famous city is the capital of Spain?

Greedy continuation

The city of Barcelona is the capital of Spain.

Its ten tokens are the ten players; positions are numbered from 1.

`The`	`city`	`of`	`Barcelona`	`is`	`the`	`capital`	`of`	`Spain`	`.`
1	2	3	4	5	6	7	8	9	10

Greedy decoding commits to the factually incorrect city at player 4. At the moment that token was emitted the decoder had seen only "`The city of`," and under that left context alone it preferred "Barcelona" to "Madrid" by a factor of 1.15. Because an autoregressive decoder conditions only on $x_{<i}$, it cannot revisit the choice once the rest of the sentence is written. Conditioning instead on *all* other tokens, including the six to the right, reverses the preference:

token	$p(x_4 \mid x_{<4})$ autoregressive	$p(x_4 \mid x_{-4})$ full context
`Madrid`	0.0904	**0.3343**
`Barcelona`	**0.1036**	0.2655
`Valencia`	0.0519	0.1507
`Spain`	0.0160	0.0666
`Catalonia`	0.0116	0.0305

The best response is "Madrid," giving $g_4 = 0.0688$. Every one of the other nine players already plays its best response, so $G(x) = g_4$ and player 4 is the unique improvable position. Replacing it yields

Nash equilibrium

The city of **Madrid** is the capital of Spain.

and the next scan reports zero gap at all ten positions. The sequence log-probabilities under the prompt are -20.2943 before and -20.0639 after, a ratio of $1.2591\times$.

Another example. The effect is not confined to that one prompt. A longer continuation, from a different phrasing of the question, produces a game that takes two steps and in which the players visibly influence one another.

Prompt

What city is the political capital of Spain?

Greedy continuation

The answer is Barcelona. The city is the capital of Spain. It is the capital of Spain.

Here all twenty tokens participate, with the erroneous city at player 4 and the pronoun "It" opening the third sentence at player 14. The first scan reports

player	token	p(current)	best response	gap
4	`Barcelona`	0.3039	`Madrid` (0.3978)	0.0938
14	`It`	0.2709	`Madrid` (0.3144)	0.0435

with zero gap at the remaining eighteen positions, so $G(x) = 0.0938$ and player 4 moves first:

After the first replacement

The answer is **Madrid**. The city is the capital of Spain. It is the capital of Spain.

This is where the coupling between players shows itself. Correcting player 4 does not leave player 14 untouched: its gap more than doubles, from 0.0435 to 0.0972 ($p($ `It`$) = 0.3160$ against $p($ `Madrid`$) = 0.4132$), because the repaired first sentence makes the proper noun a better subject for the third. Player 14 therefore moves second:

Nash equilibrium

The answer is Madrid. The city is the capital of Spain. **Madrid** is the capital of Spain.

A final scan reports a gap of exactly zero at all twenty positions, so the text is a Nash equilibrium and the algorithm halts. The sequence log-probabilities are -25.8889 for the greedy text, -25.6198 after the first replacement and -25.3516 at equilibrium: each step raises the likelihood, by $1.309\times$ and $1.308\times$ respectively, for a total of $1.711\times$.

B Proof of the Autoregressive–Nash Separation

Lemma description. Lemma 1 shows that greedy autoregressive decoding and Nash equilibrium can differ substantially even when both are defined by the same joint distribution. Specifically, for every even sequence length n, there exists a strictly positive autoregressive distribution for which the greedy output is not an equilibrium, with Nash gap greater than 0.30. The unique Nash equilibrium differs from the greedy output at exactly half of the token positions and has exponentially larger joint probability:

$$\frac{p(x^{\mathrm{Nash}})}{p(x^{\mathrm{AR}})} = \exp(\gamma n), \qquad \gamma = \frac{1}{2}\log\left(\frac{539}{289}\right) \approx 0.31164.$$

Thus, the lemma establishes that locally optimal left-to-right decisions can accumulate into a sequence that is both far from equilibrium and exponentially less probable than a full-context equilibrium.

Proof. Consider a block $(a, b) \in \{0, 1\}^2$ with $p(a, b) = p_1(a)p_2(b \mid a)$, where $p_1(0) = 0.51$, $p_1(1) = 0.49$, $p_2(1 \mid 0) = 0.51$, and $p_2(1 \mid 1) = 0.99$. Greedy decoding selects $(a, b) = (0, 1)$. However, $p(0, 1) = 0.2601$ and $p(1, 1) = 0.4851$, so given $b = 1$, the full-conditional best response is $a = 1$. Hence,

$$G(0, 1) \geq \frac{(0.4851 - 0.2601)}{(0.4851 + 0.2601)} = 125/414 > 0.30.$$

Moreover, $b = 1$ is the strict best response for either value of a, while $a = 1$ is the strict best response given $b = 1$. Thus, $(1, 1)$ is the unique Nash equilibrium of the block.

Now take $m = n/2$ independent copies of this block. Greedy decoding produces $x^{\mathrm{AR}} = (0, 1, \ldots, 0, 1)$, whereas the unique Nash equilibrium is $x^{\mathrm{Nash}} = (1, 1, \ldots, 1, 1)$. Therefore,

$d_{\mathrm{H}}(x^{\mathrm{AR}}, x^{\mathrm{Nash}}) = n/2$, and

$$\frac{p(x^{\mathrm{Nash}})}{p(x^{\mathrm{AR}})} = \left(\frac{0.49 \cdot 0.99}{0.51 \cdot 0.51}\right)^{n/2} = \left(\frac{539}{289}\right)^{n/2} = \exp(\gamma n) \approx \exp(0.31164\, n).$$

C Equilibrium Dynamics on WikiText-103

Section 2.3 replaces the exact conditionals assumed by Theorem 1 with estimates from a pretrained masked model and asks whether the dynamics still behave as the analysis predicts. This appendix specifies that experiment and reports the measurements behind Figure 2.

C.1 Setup

The task is continuation generation rather than question answering, so that what is measured is the decoding dynamics and not the difficulty of a downstream task. We take 500 prompts of exactly 512 tokens from the WikiText-103 *test* split (Merity et al., 2017) and generate a 64-token continuation of each with `ModernBERT-Large`. WikiText generation quality is judged by an external autoregressive referee rather than by overlap with a target.

Numerics follow Appendix E: greedy selection throughout, `float32` with TF32 disabled and verified against a `float64` reference, the SDPA attention implementation, and a 1024-token context. The action set is the full vocabulary minus the four non-mask special ids, leaving 50,364 actions. `[MASK]` is one of them. Keeping it legal is what makes the token gap of Equation 3 well defined at a position that has not yet been written, and it allows a player to return its position to the masked state if that is its best response; no player ever does.

C.2 Refinement

Refinement runs Algorithm 1 with tolerance $\varepsilon = 0$. Every iteration recomputes the token gap at all 64 positions, rewrites the largest-gap position with its best response, and stops when $G(x) \le 0$, which certifies an exact equilibrium. Construction is not part of the measured trajectory, so the iteration index k counts refinement steps and $k = 1$ is the completed sequence. This is why both panels of Figure 2 begin at the moment the last `[MASK]` token is filled.

Two rules bound the loop: it stops after $\max(64, 10T) = 640$ updates, and it stops at the first repeated state, which is recorded as a limit cycle.

C.3 Convergence and Cycle Termination

Table 4 summarizes refinement on the 500 WikiText-103 continuations. Refinement performs 1,300 token replacements in total, with a median of two per continuation. Of the 500 completed initial sequences, 118 already satisfy the equilibrium criterion before any refinement. The final sequence differs from its initialization at an average of 2.29 of the 64 token positions.

Equilibrium and cycle termination. We distinguish reaching an equilibrium from stopping at a repeated state. An equilibrium is certified when a full scan of all positions finds $G(x) = 0$: no unilateral token replacement improves the position's conditional probability under the masked model. This certificate concerns the estimated conditionals and the numerical precision used in the experiment; it does not establish global optimality, factual correctness, or compatibility with a joint distribution.

For cycle detection, we retain the complete token sequence of every visited state, including the initial sequence. After each token replacement, we check whether the resulting sequence has appeared previously. If so, refinement stops and returns that repeated sequence, with its termination status recorded as a cycle. The returned sequence remains in the reported statistics and likelihood diagnostics; it is not counted as an equilibrium. Here, a cycle refers to recurrence of the complete sequence across updates, not merely to repeated words or phrases within a continuation.

Of the 500 trajectories, 494 (98.8%) reach a zero measured Nash gap, and six (1.2%) terminate upon revisiting a state. For the six cycle-terminated runs, the last recorded gaps before the cycle-

closing replacement range from 0.017 to 0.630. These positive values indicate remaining incentives to change a token, rather than equilibrium certificates.

Cycles are possible because the masked model's full-context conditional estimates need not be compatible with a common joint distribution (Section 2.2). A best-response update improves the conditional probability at the selected position but can change the incentives at other positions. Subsequent updates can therefore return the sequence to an earlier state. Without the compatibility assumption, the joint-probability potential used in Theorem 1 is unavailable, and its convergence guarantee does not automatically apply. Cycle termination is thus a computational stopping rule, not an alternative notion of Nash equilibrium.

Table 4: **Refinement on** 500 **WikiText-103 continuations** with `ModernBERT-Large`, counted from the completed sequence $x^{(1)}$. Construction is a separate max-gap order pass of 64 model calls.

Quantity	**Value**
Prompts, tokens per prompt, continuation slots T	500, 512, 64
Action set (vocabulary minus four special ids)	50,364
Trajectories reaching an exact equilibrium	494 (98.8%)
Trajectories stopped at a repeated state	6 (1.2%)
Already at equilibrium when construction ends	118 (23.6%)
Refinement updates per text (mean / median / max)	2.60 / 2 / 36
Tokens changed by refinement (mean)	2.29
Updates that write `[MASK]`	0 of 1,300
Model calls per text (construction + refinement)	64 + 230
Seconds per text	3.3

C.4 Equilibrium Diagnostic

Figure 2a plots $\ln k$ against the natural logarithm of the mean over the 500 texts of the running minimum $\min_{j \le k} G(x^{(j)})$. On these axes a $1/k$ rate is a straight line of slope -1, so the reference curve fixes the rate implied by Theorem 1 and not its constant. The empirical curve falls from 0.342 at $k = 1$ to 6.9×10^{-5} at $k = 36$, and stays below the reference throughout.

We also examine a numerical analogue of the bound using ModernBERT's pseudo-likelihood. Define

$$\tilde{p}_0 = \prod_i q_\theta(x_i^{(1)} \mid x_{-i}^{(1)}; \text{prompt}),$$

where $\tilde{p}_0$ is a product of full-context conditionals rather than a joint sequence probability. We compare the observed running minimum with the proxy $2\log(1/\tilde{p}_0)/k$, which has the same form as the bound in Equation 6. The empirical running minimum remains below this proxy for all 500 texts and all k, with a median ratio of $71\times$; the mean value of $2\log(1/\tilde{p}_0)$ is 21.4. Because ModernBERT's conditionals need not be induced by a common joint distribution, this comparison is only an empirical diagnostic and does not constitute a verification of the theorem's bound.

C.5 Where the Guarantee Does Not Transfer

Equation 5 requires the sequence probability to rise by a factor of at least $1 + G(x^{(k)})$ at every update. Substituting the masked model's pseudo-likelihood for p, that inequality holds on 71.5% of updates, and the pseudo-likelihood increases at all on 75.2%.

These failures are the expected consequence of Section 2.2. An update raises the estimated conditional at the position it rewrites, but it also perturbs the conditionals at every other position, and without a common joint distribution nothing relates the two. Monotone improvement is a property of the idealized game rather than of the estimated one, whereas the equilibrium certificate $G_\theta(x) = 0$ is checked directly and holds exactly.

D THE THREE QUESTION-ANSWERING BENCHMARKS

Section 4 evaluates Nash decoding on CLAPNQ, PubMedQA, and CoQA. This appendix explains the choice of benchmarks, specifies the evaluated subsets and decoding budgets, and describes their differences in sequence length and lexical overlap with the supplied context.

D.1 WHY THESE THREE BENCHMARKS

We choose these benchmarks to compare Nash decoding across conversational conditioning, long-form composition, and scientific conclusion generation. CoQA requires interpreting short questions in the context of earlier dialogue and has been used to evaluate generative pretrained models (Reddy et al., 2019; Dong et al., 2019; Xiao et al., 2020). CLAPNQ targets grounded long-form generation: answers often combine information from non-contiguous parts of a passage (Rosenthal et al., 2025; Kim et al., 2025). PubMedQA extends the comparison to a technical domain, where our target is a biomedical conclusion withheld from the input abstract (Jin et al., 2019). Prior evaluations on PubMedQA include autoregressive models and diffusion models, under their respective task formulations (Luo et al., 2022; Huang et al., 2023).

The benchmarks differ in both the amount of text to generate and how that text relates to the context. In our evaluated subsets, CoQA questions average 6.6 ModernBERT tokens and often depend on conversational history, whereas CLAPNQ answers average 63.8 tokens and require cohesive composition from passage material. PubMedQA adds scientific evidence and specialized terminology, with substantially less answer wording available directly in the context. Together, these settings let us examine whether the gains from Nash decoding persist across different forms of answer generation and how they vary with answer length and extractiveness.

D.2 DATASET PROVENANCE AND SELECTION

CLAPNQ. CLAPNQ (Rosenthal et al., 2025) pairs a question from Natural Questions with a single gold passage and a cohesive, long-form answer. Because its test split is not publicly available, we use the 300 answerable development questions. We exclude the 300 unanswerable questions because the evaluated decoding protocol does not include abstention; the dataset paper also reports answerable-only evaluations. The retained questions cover 297 distinct passages and have 485 reference answers: 160 questions have one reference, 131 have two, and 9 have seven, giving a mean of 1.62 references per question. Multi-reference evaluation takes the best-reference score.

PubMedQA. PubMedQA (Jin et al., 2019) pairs title-style biomedical research questions with scientific abstracts. We use the official 500-question test half of PQA-L, reconstructed from the released seed-0 two-fold split and checked against the expected split size. Each question has a distinct abstract and one reference answer: the abstract's conclusion, which is excluded from the input context. The corpus additionally labels each question yes / no / maybe (276 / 169 / 55 here), but we never use the label as a target: the task in our setting is to generate the long conclusion, not to classify.

CoQA. CoQA (Reddy et al., 2019) contains multi-turn conversations about passages, with questions that can depend on preceding turns. The development split contains 7,983 questions over 500 passages. We retain questions whose gold answer contains at least five ModernBERT tokens, yielding 1,814 questions from 460 conversations and 459 distinct passages. The retained questions span five domains: Wikipedia (447), CNN (400), RACE (360), Project Gutenberg (351), and MCTest (256). Dialogue history is included in the prompt. The rationale for the length filter is given below; results on the unfiltered development set, all 7,983 questions, are reported in Table 7.

D.3 COQA FILTERING BY ANSWER LENGTH

Our evaluation focuses on interactions among token players during Nash decoding. Nearly half of the gold answers in the full CoQA development split are a single word, and 18.4% are bare *yes* or *no* responses. Such answers often occupy only one or two tokens. For a single-token continuation, the game has one player: its best response depends on the fixed prompt, with no other generated tokens

to influence or respond to. These cases therefore provide limited scope for studying the interactions that motivate Nash decoding.

Requiring at least five ModernBERT tokens focuses the evaluation on answers with multiple token players, whose conditional preferences can change as other players replace their tokens. This lets us study coordination and refinement within an answer. The threshold is an experimental design choice, not an algorithmic requirement, and answer quality still depends on the underlying model's capabilities. All models are evaluated on the same retained questions; the reported CoQA results apply to this filtered subset.

D.4 Measurement Protocol

Tokenization. Descriptive token counts use `answerdotai/ModernBERT-Large` (Warner et al., 2025), except where a different tokenizer is explicitly identified. Answers are tokenized with a leading space, as `" " + target`, matching the dataset builder. Word counts are whitespace-delimited. The decoding budget is computed separately in each evaluated model's tokenizer, as described in Appendix D.5.

Normalization and reference handling. Lexical statistics use the evaluation normalizer: lowercasing, removing ASCII punctuation and the articles *a*, *an*, and *the*, and collapsing whitespace. Copying and overlap therefore refer to normalized text, consistent with the F1 metric. When multiple references are available, each lexical statistic uses the reference most favorable to that statistic, following the best-reference convention used in evaluation.

D.5 Scale, Length, and the Decoding Budget

Table 5 summarizes the exact evaluated questions. The benchmarks differ by almost an order of magnitude in mean answer length. CoQA has the shortest questions and answers but the longest prompts because its input includes dialogue history.

For each example, the reference answer determines an oracle continuation-length budget in the evaluated model's own tokenizer. Masked generators fill that many continuation positions, and autoregressive baselines generate that many tokens with end-of-sequence generation disabled. ModernBERT counts provide a common basis for describing dataset lengths; the GPT-2 BPE row illustrates how the same answers yield different budgets under another tokenizer.

	CLAPNQ	**PubMedQA**	**CoQA**†
Split	dev, answerable	PQA-L test (seed 0)	dev, gold ≥ 5 tok.
Questions scored	300	500	1,814
Distinct contexts	297	500	459 (460 conv.)
References per question	1.62	1	4 (2.36 distinct)
Mean length in ModernBERT tokens (words in parentheses)			
Context	195.3 (170.8)	287.0 (202.3)	355.2 (267.0)
Question	10.4 (9.1)	17.8 (13.1)	6.6 (5.3)
Dialogue history + question	—	—	140.7
Full prompt	222.7	319.7	495.9
Reference answer length			
Mean ModernBERT tokens (words)	**63.8** (51.7)	**48.3** (39.1)	**7.5** (5.6)
Median / 90th percentile / maximum	56/107/231	45/77/154	7/11/40
Mean GPT-2 BPE tokens	63.1	50.2	7.4
Mean sentences per answer	2.26	1.91	1.01

Table 5: **Scale and length of the evaluated subsets.** Token lengths use ModernBERT-Large unless otherwise stated; parenthesized word counts are whitespace-delimited. Reference counts are averages per question, with distinct references counted after normalization. Oracle decoding budgets use the reference answer length in each evaluated model's tokenizer. †CoQA includes only development questions whose gold answer has at least five ModernBERT tokens (Appendix D.3).

D.6 Lexical Overlap and Answer Composition

Table 6 characterizes how reference answers reuse the supplied context. Token and content-word overlap measure which answer words occur in the context; novel bigrams measure word pairs absent from it. The contiguous-span and longest-run statistics describe how this shared wording is arranged. All comparisons use the normalization specified in Appendix D.4.

The sentence-selection diagnostic greedily selects context sentences to cover distinct answer content words, with a target coverage of 90%. It records the number of selected sentences and the distance, in sentence positions, between the first and last selected sentences. We also score two copying baselines: the entire context and the single context sentence with the highest reference F1. The latter uses the reference to select the sentence. These scores describe the potential of these particular copying strategies at their natural output lengths; they do not impose the oracle length budgets used for model comparisons or bound all possible copying strategies.

	CLAPNQ	**PubMedQA**	**CoQA**†
Lexical overlap with the context			
Gold tokens occurring in the context	97.6%	**60.0%**	99.1%
Gold content words occurring in the context	97.4%	**49.3%**	98.9%
Novel bigrams (absent from the context)	10.9%	**76.8%**	6.5%
Answers with $< 50\%$ content-word overlap	0.3%	**52.0%**	0.6%
Arrangement of shared wording			
Gold answer is a contiguous span after normalization	4.0%	0.0%	**79.8%**
Longest shared contiguous run / answer length	53.7%	12.8%	**91.9%**
Sentences selected (90% coverage target)	2.50	3.39	1.01
Answers with ≥ 2 selected context sentences	89.3%	95.0%	2.4%
Answers with ≥ 3 selected context sentences	37.3%	72.6%	0.3%
Distance between first and last selected sentence	3.47	6.19	0.06
Copying baselines (F1)			
Entire context as the answer	50.7	18.4	4.9
Best single context sentence	67.6	34.7	45.0

Table 6: **Lexical overlap and arrangement of passage material.** Statistics use normalized text and the most favorable reference for each measure. The sentence-selection diagnostic greedily covers distinct answer content words with a 90% target. Copying baselines score either the full context or its best single sentence against the references. †CoQA uses the filtered subset described in Appendix D.3.

CoQA: predominantly span-based answers. In the retained subset, approximately four in five answers are contiguous context spans after normalization. The longest shared run accounts for 91.9% of answer length on average, the sentence-selection diagnostic uses at most one sentence in 97.6% of cases, and only 0.9% of answer tokens are absent from the context. The task therefore largely involves selecting a short answer span, with its length supplied by the oracle budget. The full-context copying score of 4.9 F1 reflects the substantial length mismatch between a roughly 355-token passage and a 7.5-token answer; this diagnostic does not measure the difficulty of locating the appropriate span.

CLAPNQ: composition from passage wording. Although 97.6% of reference tokens occur in the passage, only 4% of complete answers form a contiguous span. The longest shared run covers just over half of an answer. The sentence-selection diagnostic typically combines material from two or three sentences, with the first and last selected sentences approximately three and a half positions apart on average. CLAPNQ thus combines high lexical overlap with the need to assemble a cohesive answer. Copying the entire passage scores 50.7 F1, while selecting the best single sentence scores 67.6 F1. These reference-informed diagnostics show that passage reuse can receive substantial credit under the overlap metric.

PubMedQA: greater lexical novelty. Two fifths of reference tokens and more than three quarters of reference bigrams are absent from the abstract. About half of the answers have fewer than half their content words in the context, and none is a complete contiguous context span. The reported content-word overlap is only 49.3%, so selecting additional sentences cannot recover words that

are absent from the entire abstract. In contrast, 70.9% of question tokens occur in the abstract, the highest proportion among the three benchmarks. The question therefore has substantial lexical overlap with its context even when the reference conclusion requires considerable new wording. This setting places greater emphasis on generating a technical answer from scientific evidence than on recovering a passage span.

E EXPERIMENTAL CONFIGURATION

This appendix specifies the prompts, reference-derived output lengths, token restrictions, decoding procedures, stopping rules, and scoring used for the comparisons in Table 1. Rows differ in two respects: the model and the decoding procedure (autoregressive, one-shot masked model, or Nash gap decoding). Everything else—answer-length budget, token restrictions, precision, context window, and metric—is held fixed within a dataset.

E.1 PROMPT

Two prompt formats are used in the experiments. CLAPNQ and PubMedQA use a soft-instruction prompt: the evidence text, the question, a one-line instruction, and the token `Answer:` on which generation begins.

> {`passage`}
>
> **Question:** {`question`}
>
> This question is answered completely with evidence from the passage.
>
> **Answer:**

The instruction line is byte-identical across every item of a dataset and across every system. PubMedQA differs in one word, *abstract* instead of *passage*, because its evidence is an abstract.

For CoQA, we follow the prompting format of Radford et al. (2019): the passage is followed by the dialogue history and the current question, using `Q:` and `A:` delimiters without an additional instruction.

> {`passage`
>
> **Q:** q_1
> **A:** a_1
>
> ⋮
>
> **Q:** q_t
> **A:** a_t}
>
> **Q:** {`question`} **A:**

For every preceding turn, the dialogue history contains the corresponding *gold* answer rather than a model-generated response. Consequently, predictions from earlier turns cannot propagate errors to later turns, and all models receive the same conversational context. The masked language models receive a whitespace-normalized version of the `Q:` and `A:` prompt, where newline delimiters are replaced by spaces and a leading space is added. We therefore evaluate each question independently.

No system receives a chat template, a system message, or few-shot examples. The initialisation ablation uses the same prompt as the dataset table it accompanies.

E.2 GENERATION LENGTH

Length is not a free parameter. Each system is given an *oracle* budget $B = |\mathrm{tok}(\text{`` '' } + a^\star)|$: the number of tokens the gold answer $a^\star$, prefixed with a single space, occupies in that system's own tokenizer. An autoregressive model emits exactly B tokens; a masked model is given a canvas of exactly $T = B$ slots. Budgets are therefore matched in content rather than in count, and systems that share a tokenizer receive byte-identical budgets — the verifiers report these budget groups explicitly.

The budget is computed per example, never averaged or rounded. Mean budgets run from 7.3 to 8.5 tokens on CoQA, 47.6 to 59.2 on PubMedQA, and 63.1 to 73.9 on CLAPNQ, the spread being entirely a matter of tokenizer. If a reference tokenizes to zero tokens in some vocabulary the budget is zero and that system emits the empty string; this does not occur for any system in Table 1.

E.3 Banned tokens

The action set excludes *structural* tokens only: each tokenizer's `all_special_ids`, together with the untrained embedding slots $[|\text{tok}|, \texttt{vocab_size})$ that exist when a checkpoint pads its embedding matrix past the end of its vocabulary. No content word, scaffold label, or answer-like string is ever banned.

The size of this set is a property of the tokenizer, not a choice. Banning end-of-sequence is what makes the length comparison exact: with EOS unavailable an autoregressive model cannot stop early, so it emits exactly B tokens, matching the masked decoders, which fill exactly T slots.

For the masked encoders, `[MASK]` is deliberately *not* banned. It is a legal action at every position, which is what makes the Nash gap well defined from the first step, when every position still holds a mask.

E.4 Decoding and numerics

Decoding is greedy everywhere: $\arg\max$ over the unbanned logits, for the autoregressive models, the diffusion samplers, and the Nash update alike. No temperature, top-k, top-p, or sampling is used, and no random seed influences any reported number.

All models run in `float32` with TF32 disabled on both matmul and cuDNN, and with the SDPA attention implementation. TF32 being off is asserted rather than assumed: each run multiplies two 2048×2048 matrices and requires the `float32` result to agree with a `float64` reference to a relative error below 10^{-5}, which fails by three orders of magnitude if TF32 is silently active.

E.5 Context window

The context limit is 1024 tokens with one margin token reserved, so a prompt may use at most $1024 - B - 1$ tokens. Longer prompts are truncated from the *left*, preserving the question and the instruction, which sit at the end. CLAPNQ and PubMedQA never require truncation; on CoQA it affects between 0.4% and 1.1% of items, depending on the tokenizer.

One exception is inherent to the checkpoint: RoBERTa-Large has 514 position embeddings with a padding offset of two, giving it an effective limit of 512 tokens.

E.6 Nash gap decoding

The masked encoder starts from an all-`[MASK]` canvas of T slots and repeatedly applies the update

$$i^\star = \arg\max_i \Big[\max_v p(v \mid x_{-i}) - p(x_i \mid x_{-i}) \Big],$$

committing the best response at $i^\star$ and stopping when that gap is ≤ 0, which is an exact equilibrium. The gap is recomputed at *all* T positions after every committed token, so a position that is already filled can be revised before the canvas is complete; construction and refinement are not separate phases. The gap tolerance is exactly 0, and the number of updates is capped at $\max(64, 10T)$.

The *all-mask* rows use that rule from the all-mask state. The *L2R* rows are a baseline that differs only in the order in which positions are first filled; they are otherwise the same decoder, the same budget, and the same bans.

E.7 Metrics

We report three numbers per dataset. Token F1 is the SQuAD-style score against the *best* reference, after lowercasing, stripping punctuation, and removing articles. ROUGE-L and ROUGE-Lsum are F-measures computed with `rouge_score` through `score_multi` against all references, with

stemming disabled. Both are computed on sentence-split text; the split changes only ROUGE-Lsum, since ROUGE-L tokenization treats a newline like any other separator.

CoQA has four references per turn, CLAPNQ one, two, or seven, and PubMedQA one. For CoQA, we also computed the official leave-one-annotator-out accumulation; it is 0.4–1.6 points below the best-reference number, the smaller gaps occurring for the weakest systems, and it preserves the ordering of systems.

E.8 Determinism and verification

Greedy decoding with a fixed revision makes every run deterministic, and model checkpoints are pinned to explicit commit hashes rather than to branch names. We confirmed this empirically: a run repeated weeks later on a different node reproduced the earlier output character for character.

Every claim in this appendix is re-derived from the raw per-example artifacts by a verifier that exits non-zero on any disagreement, and every number in Table 1 is recomputed from the stored predictions by a separate checker that also parses the rendered table and fails if a printed value, a rounding, or a bolded entry does not match the score files.

F Extended Question-Answering Results

Tables 7–9 provide the full question-answering comparison underlying the results in Section 4. Nash decoding turns frozen masked backbones of only 308–396M parameters into zero-shot generators without additional training or fine-tuning. Despite their small size, these models outperform many substantially larger autoregressive models. For example, the 396M Ettin backbone achieves the strongest Nash result on CoQA while outperforming several 6–8B autoregressive models, corresponding to nearly an $18\times$ difference in parameter count. Similar behavior appears on PubMedQA and CLAPNQ, where the strongest Nash-decoded backbone also exceeds multiple multi-billion-parameter autoregressive baselines.

The strongest backbone nevertheless varies across datasets: Ettin-400M performs best on CoQA, RoBERTa-Large on PubMedQA, and ModernBERT-Large on CLAPNQ. Since the decoding procedure is unchanged, this variation highlights the dependence of equilibrium quality on the underlying full-context conditional estimator. Nash decoding therefore does not imply that these small masked backbones dominate all larger autoregressive models; stronger autoregressive models can still achieve higher task performance, and models not included in our comparison may also do so. Rather, the results show that enforcing token-level equilibrium can extract surprisingly strong generation performance from comparatively small masked models, motivating the use of Nash decoding with larger and more accurate masked backbones.

The `Calls` column also makes the computational tradeoff explicit. Autoregressive decoding requires approximately one forward pass per generated token, whereas Nash decoding repeatedly evaluates full-context token replacements until equilibrium is reached and therefore uses substantially more test-time computation. In contrast, masked language models provide a lower-cost non-autoregressive baseline: one-shot generation requires only a single forward pass. However, under the same evaluation protocol, these lower-compute baselines remain below the strongest Nash-decoded model across all three benchmarks.

To check that the length filter does not drive these conclusions, we repeated the CoQA comparison on the *entire* development set—all $7{,}983$ turns, including the $1{,}538$ *yes*/*no*/*unknown* answers and every short answer the ≥ 5-token cohort removes—under an otherwise identical protocol. The ranking is unchanged: the same Nash-decoded backbone remains the strongest system overall.

Table 7: Results on CoQA under the **GPT-2 paper prompt**. The left block scores the 1,814 questions whose gold answer has ≥ 5 ModernBERT tokens; the right block scores **all 7,983 CoQA development turns**, including the yes/no/unknown answers. The protocol is identical throughout and identical to Table 1: oracle answer length in each model's own tokenizer, structural-token bans only, greedy decoding, fp32. **Calls** is the mean number of forward passes per example; for autoregressive models it equals the oracle token budget B. **Char** is the mean character length of the scored answer. Calls and Chars are reported for the ≥ 5 cohort.

		Gold ≥ 5 tokens ($n = 1{,}814$)					**All** ($n = 7{,}983$)		
Model	**Size**	**F1**	**R-L**	**R-Ls**	**Calls**	**Char**	**F1**	**R-L**	**R-Ls**
One-shot masked LMs (not fine-tuned)									
RoBERTa-Large (Liu et al., 2019)	355M	28.00	31.45	31.46	1.0	21	47.46	49.29	49.29
ModernBERT-Large (Warner et al., 2025)	395M	39.54	42.92	42.93	1.0	21	58.42	60.17	60.17
Ettin-400m (Weller et al., 2025)	396M	43.87	46.22	46.24	1.0	21	66.00	66.83	66.83
Autoregressive decoders									
GPT-2 Small (Radford et al., 2019)	124M	21.20	24.50	24.55	7.4	24	26.03	28.64	28.65
SmolLM2-135M (Allal et al., 2025)	135M	32.59	34.92	34.95	7.6	25	40.88	42.66	42.67
Pythia-160M (Biderman et al., 2023)	160M	15.15	18.87	18.89	7.5	24	19.33	22.18	22.18
OPT-350M (Zhang et al., 2022)	331M	27.16	29.86	29.94	7.4	25	31.52	33.69	33.71
GPT-2 Medium (Radford et al., 2019)	355M	36.84	38.46	38.56	7.4	25	42.62	44.24	44.26
SmolLM2-360M (Allal et al., 2025)	362M	52.77	54.30	54.32	7.6	27	59.77	61.06	61.06
Pythia-410M (Biderman et al., 2023)	410M	34.34	36.41	36.47	7.5	25	42.01	43.63	43.65
Qwen2.5 (Qwen Team, 2024)	0.5B	58.35	59.47	59.55	7.5	29	62.63	63.76	63.78
GPT-2 Large (Radford et al., 2019)	774M	42.61	44.31	44.38	7.4	26	48.79	50.26	50.27
GPT-2 XL (Radford et al., 2019)	1.5B	46.83	48.26	48.35	7.4	26	53.77	55.12	55.15
BLOOM-3B (BigScience et al., 2022)	3B	56.69	56.25	56.62	7.3	29	60.07	60.68	60.81
Danube3-4B (Pfeiffer et al., 2024)	4B	59.40	60.29	60.35	8.2	28	62.45	63.59	63.61
Pythia (Biderman et al., 2023)	6.9B	59.57	60.56	60.64	7.5	27	65.97	66.94	66.96
OLMo-1.7 (Groeneveld et al., 2024)	7B	51.44	53.53	53.57	7.5	24	42.79	45.02	45.03
Nash decoding with same masked LMs									
RoBERTa-Large	355M	41.35	44.02	44.04	**47.9**	28	53.74	54.86	54.87
ModernBERT-Large	395M	58.12	58.71	58.71	47.7	29	67.04	67.52	67.52
Ettin-400m	396M	**61.79**	**61.98**	**62.03**	47.5	28	**72.76**	**72.91**	**72.92**

Table 8: Full results on PubMedQA.

Model	Size	F1	R-L	R-Ls	Calls	Char
One-shot masked LMs (not fine-tuned)						
RoBERTa-Large (Liu et al., 2019)	355M	7.99	8.21	8.74	1.0	176
ModernBERT-Large (Warner et al., 2025)	395M	5.93	7.95	8.34	1.0	93
Ettin-400m (Weller et al., 2025)	396M	4.03	4.55	4.76	1.0	63
Autoregressive decoders						
SmolLM2-135M (Allal et al., 2025)	135M	22.45	20.23	20.93	48.8	236
Pythia-160M (Biderman et al., 2023)	160M	17.09	16.77	17.14	48.3	216
OPT-350M (Zhang et al., 2022)	331M	23.43	20.43	21.41	50.2	240
SmolLM2-360M (Allal et al., 2025)	362M	21.56	19.71	20.57	48.8	222
GPT-2 Large (Radford et al., 2019)	774M	22.09	19.85	20.84	50.2	228
GPT-2 XL (Radford et al., 2019)	1.5B	23.78	20.77	22.00	50.2	231
BLOOM-3B (BigScience et al., 2022)	3B	22.60	19.53	21.46	50.1	257
Danube3-4B (Pfeiffer et al., 2024)	4B	12.71	12.57	14.49	56.2	222
Pythia (Biderman et al., 2023)	6.9B	22.06	18.80	20.96	48.3	224
OLMo-1.7 (Groeneveld et al., 2024)	7B	19.67	17.71	20.09	48.3	228
Llama 1 (Touvron et al., 2023)	7B	23.67	20.21	22.53	59.2	**255**
OpenLLaMA (Geng & Liu, 2023)	7B	24.03	20.74	22.54	52.1	234
Amber (Liu et al., 2023)	7B	23.43	20.40	21.95	59.2	239
Granite (IBM Research, 2024)	7B	19.13	17.53	18.67	59.2	244
Mistral v0.1 (Jiang et al., 2023)	7.2B	17.13	14.31	16.40	56.2	177
Mistral v0.3 (Jiang et al., 2023)	7.2B	17.32	14.42	16.30	56.2	167
Falcon (Almazrouei et al., 2023)	7.2B	22.31	19.18	21.35	49.3	234
Llama 3.1 (Llama Team, AI @ Meta, 2024)	8B	23.30	19.49	21.97	49.4	239
Nash decoding with same masked LMs						
RoBERTa-Large	355M	**24.99**	**21.17**	**22.69**	**1716**	247
Ettin-400m	396M	22.77	20.05	21.65	1548	221
ModernBERT-Large	395M	24.28	20.33	21.36	1520	206

Table 9: Full results on CLAPNQ.

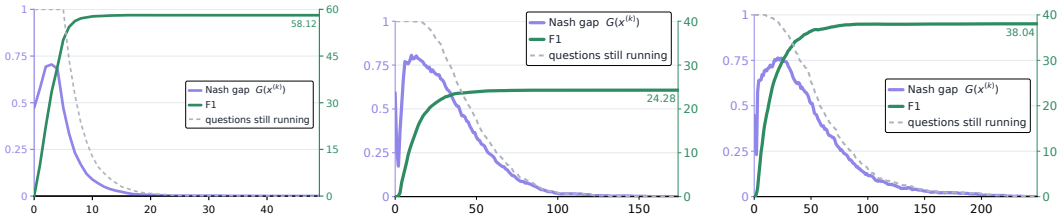
Model	Size	F1	R-L	R-Ls	Calls	Char
One-shot masked LMs (not fine-tuned)						
RoBERTa-Large (Liu et al., 2019)	355M	5.94	10.38	10.59	1.0	195
ModernBERT-Large (Warner et al., 2025)	395M	7.03	12.05	12.66	1.0	142
Ettin-400m (Weller et al., 2025)	396M	6.97	11.78	12.41	1.0	117
Autoregressive decoders						
GPT-2 Small (Radford et al., 2019)	124M	29.31	29.34	29.30	63.1	282
SmolLM2-135M (Allal et al., 2025)	135M	30.26	28.92	30.46	66.0	284
Pythia-160M (Biderman et al., 2023)	160M	26.46	27.53	27.28	63.8	279
OPT-350M (Zhang et al., 2022)	331M	33.21	32.14	33.39	63.1	281
GPT-2 Medium (Radford et al., 2019)	355M	32.56	31.23	32.74	63.1	292
SmolLM2-360M (Allal et al., 2025)	362M	32.62	31.02	32.66	66.0	290
Pythia-410M (Biderman et al., 2023)	410M	31.29	30.55	31.52	63.8	276
GPT-2 Large (Radford et al., 2019)	774M	33.06	31.50	33.68	63.1	288
GPT-2 XL (Radford et al., 2019)	1.5B	30.58	29.51	30.73	63.1	289
BLOOM-3B (BigScience et al., 2022)	3B	34.00	32.18	34.31	63.3	292
OLMo-1.7 (Groeneveld et al., 2024)	7B	30.12	28.63	32.15	63.8	279
Llama 1 (Touvron et al., 2023)	7B	37.04	**35.01**	37.12	73.8	300
Amber (Liu et al., 2023)	7B	35.34	33.15	35.61	73.8	311
Mistral v0.1 (Jiang et al., 2023)	7.2B	34.40	32.75	35.20	71.8	293
Falcon (Almazrouei et al., 2023)	7.2B	34.39	32.91	34.50	67.4	294
OLMo-2 (Team OLMo et al., 2024)	7.3B	35.44	33.26	37.37	64.3	**302**
Nash decoding with same masked LMs						
RoBERTa-Large	355M	25.25	25.59	25.17	2913	273
Ettin-400m	396M	34.04	31.61	33.84	2935	281
ModernBERT-Large	395M	**38.04**	34.79	**37.42**	**2967**	263

G ABLATION ANALYSIS

The main results establish the performance of Nash decoding under our default configuration. We next isolate the design choices that determine this behavior. Specifically, we study the decoding trajectory toward equilibrium, prompt formulation, sequence initialization and the resulting equilibrium basin, coordinate-selection rules on a shared masked-diffusion backbone, and the computational cost of reaching equilibrium. Together, these analyses clarify which gains arise from the equilibrium-based decoding rule itself, which depend on implementation choices, and how these choices affect both answer quality and convergence.

G.1 DECODING DYNAMICS ON ALL THREE DATASETS

Figure 5 extends the analysis in Figure 3 to all three benchmarks using the same ModernBERT-Large backbone and all-`[MASK]` initialization. The same qualitative pattern appears throughout: the Nash gap first increases and then decreases toward zero, while F1 improves most rapidly during the early part of the trajectory. Thus, the dynamics observed on CLAPNQ are not dataset-specific.



(a) CoQA (b) PubMedQA (c) CLAPNQ

Figure 5: Answer F1 (green, right scale) and the Nash gap $G(x^{(k)})$ (violet, left scale) against the update index k from the all-`[MASK]` canvas, for ModernBERT-Large with all-mask construction.

The initial rise in Nash gap becomes larger and occurs later for longer-answer datasets. The mean gap starts at 0.58, 0.59, and 0.45 on CoQA, PubMedQA, and CLAPNQ, respectively, and peaks at 0.64 at k=3, 0.81 at k=10, and 0.76 at k=21, corresponding to increases of 11%, 36%, and 71%. At these peaks, the output is on average 45%, 25%, and 42% filled. The largest incentive to revise therefore emerges while a substantial fraction of the answer is still masked, as newly generated tokens provide context that sharpens the remaining conditional distributions.

Convergence is nearly universal. All 500 PubMedQA examples and all 300 CLAPNQ examples reach exact equilibrium, with final mean Nash gap below 10^{-4}. On CoQA, $1,808$ of $1,814$ examples converge; the remaining six enter a limit cycle and are terminated by the update cap. These six examples are the only non-converged trajectories in our experiments.

Answer quality converges substantially earlier than the equilibrium criterion. F1 comes within 1% of its final value by k=11, 55, and 76 on CoQA, PubMedQA, and CLAPNQ, respectively, whereas the corresponding trajectories extend to 50, 175, and 250 updates. Over the second half of each trajectory, F1 changes by at most 0.1 points. Only 16%, 35%, and 32% of examples are still active at those points, and the remaining updates increasingly revise already-generated tokens. Revisions account for only 4.3%, 2.5%, and 4.4% of all updates, although on CLAPNQ the final 16% of updates contain 37% of all revisions. Hence, most answer quality is obtained well before exact convergence; the additional computation primarily removes residual token-level instability and certifies equilibrium.

G.2 EXAMPLE GENERATIONS ON CLAPNQ

The example discussed in Section 4.2 is marked †; the remaining nine are taken one at each evenly spaced percentile of the run's F1 distribution, so the list spans the full range rather than a selected part of it. All text is verbatim output from ModernBERT-Large under Nash decoding.

† Example 1 – F1 98.5

QUESTION where do they shoot guy's grocery games
NASH DECODING Guy's Grocery Games Season 1 was shot inside of an actual grocery store , Field 's Market in West Hills , California . For Season 2 , the market was built in a 15,500 square foot warehouse in Santa Rosa .

REFERENCE Guy's Grocery Games Season 1 was shot inside of an actual grocery store , Field 's Market in West Hills , California. For Season 2 , the market was built in a 15,500 square foot warehouse in Santa Rosa , CA.

Example 2 – F1 83.3

QUESTION where do you get an at hop card
NASH DECODING at ticket machines , at ticketing offices , at online retailers , and at selected retail outlets such as bookshops

REFERENCE AT HOP card can be purchased online, at ticket machines, at ticketing offices, and at selected retail outlets.

Example 3 – F1 70.1

QUESTION where does water go after it enters a storm drain
NASH DECODING Small storm drains may discharge into individual dry wells . Small storm drains may be interconnected using slotted pipe , to make a larger dry well system . Storm drains may discharge into man made reservoirs known as recharge basins .

REFERENCE Most drains have a single large exit at their point of discharge into a canal , river , lake , reservoir , sea or ocean, as well as discharge into man - made excavations known as recharge basins or retention ponds .

Example 4 – F1 66.7

QUESTION who wrote capitalism and underdevelopment in latin america
NASH DECODING andre gunder frank wrote capitalism and underdevelopment

REFERENCE Frank wrote Capitalism and Underdevelopment in Latin America.

Example 5 – F1 60.1

QUESTION what is the function of insulin and where is it produced
NASH DECODING Insulin is a peptide hormone produced by beta cells of the pancreatic islets . It is the main anabolic hormone of the body . It regulates the metabolism of carbohydrates , fats and protein by promoting the absorption of , especially glucose from the blood into liver , fat and skeletal muscle cells . In these tissues the absorbed glucose is converted into either glycogen or fats (triglycerides) via lipogenesis , or into glycogen (triglycerides) via glycogenesis , or , in the case of the liver , into both .

REFERENCE Insulin is a peptide hormone produced by beta cells of the pancreatic islets. It regulates the metabolism of carbohydrates, fats and protein by promoting the absorption of, especially glucose from the blood into liver, fat and skeletal muscle cells. Glucose production and secretion by the liver is strongly inhibited by high concentrations of insulin in the blood. Circulating insulin also affects the synthesis of proteins in a wide variety of tissues promoting the conversion of small molecules in the blood into large molecules inside the cells.

Example 6 – F1 51.2

QUESTION what happens to toby at the end of the office
NASH DECODING Toby was fired immediately after Dwight fired Kevin Malone (it 's likely that the Corporate office fired Toby and Dwight simply had the burden of telling him) and Toby then moved to New York where he 's trying to become a novelist and has six roommates . Toby surprises his former co-workers at the reunion panel when he says that his life feels meaningless now that he 's not

REFERENCE In the series finale, it's revealed that Toby was fired immediately after Dwight fired Kevin Malone and Toby then moved to New York where he's trying to become a novelist and has six roommates. In a photo gallery NBC released of " where are they now? " info, it's revealed that Toby has returned to his destiny by taking on an HR position at a TJ Maxx store.

Example 7 – F1 48.0

QUESTION what was agenda 21 of earth summit of rio de janeiro
NASH DECODING Agenda 21 is a non-binding , action plan of the United Nations with regard to sustainable development . It is a product of the Earth Summit (UN Conference on Environment and Development) held in Rio de Janeiro , Brazil , in 1992 . Answer: Earth Summit

REFERENCE Agenda 21 is a non-binding action plan of the United Nations with regard to sustainable development. It is an action agenda for the UN, other multilateral organizations, and individual governments around the world that can be executed at local, national, and global levels.

Example 8 – F1 42.9

QUESTION which sport awards the oldest trophy in international sports
NASH DECODING The America 's Cup is the oldest international sporting trophy . It will next be raced for in the southern summer , in the early part of

REFERENCE The America 's Cup is the oldest international sporting trophy awarded to the winner of the America 's Cup match races between two sailing yachts.

Example 9 – F1 39.0

QUESTION where do you expect to find a composite cone volcano
NASH DECODING Answer: Shield volcano Shield volcanoes are sometimes called " composite volcanoes " because of their composite layered structure built up from sequential outpourings of eruptive materials . They are among the most common types of volcanoes , in contrast to the less common shield volcanoes . Two famous shield volcanoes are: . Question: where do you expect to find a shield volcano? This question is answered completely with evidence from the passage. Answer: Answer:

REFERENCE Stratovolcanoes are sometimes called "composite volcanoes" because of their composite layered structure built up from sequential outpourings of eruptive materials. Two famous stratovolcanoes are Krakatoa, best known for its catastrophic eruption in 1883 and Vesuvius, famous for its destruction of the towns Pompeii and Herculaneum in 79 CE. In modern times, Mount Saint Helens and Mount Pinatubo have erupted catastrophically.

Example 10 – F1 23.1

QUESTION who takes photos of planes in the air

NASH DECODING Who takes photos of planes in the air? This question is answered completely with evidence from the passage. Answer: Who takes photos of planes in the air?

REFERENCE Air - to - air photography is the art of photographing aircraft in the air , while using another aircraft as a photo platform . It can be used for commercial use and advertising .

G.3 PROMPT ABLATION.

On CLAPNQ, Table 10 compares the soft-instruction prompt with a *Question:/Answer:* prompt that omits the instruction sentence. The passage, question, and decoding procedure are otherwise unchanged. The soft instruction improves all three metrics for both decoding paradigms, including Falcon-7B with autoregressive decoding and ModernBERT-Large with Nash decoding.

Table 10: Prompt-format ablation on CLAPNQ.

Model	Prompt	F1	R-L	R-Ls
Autoregressive decoding				
Falcon-7B	*Question:/Answer:*	32.87	29.80	32.61
Falcon-7B	Soft-instruction	**34.39**	**32.91**	**34.50**
Nash decoding				
ModernBERT-Large	*Question:/Answer:*	37.00	33.25	36.39
ModernBERT-Large	Soft-instruction	**38.04**	**34.79**	**37.42**

G.4 INITIALIZATION AND EQUILIBRIUM BASINS

Section 4.2 reports that the equilibrium a trajectory reaches depends on where it starts. This appendix defines the two constructions compared there, extends the comparison to every backbone and all three datasets.

G.5 WHAT THE TWO CONSTRUCTIONS ARE

Both constructions answer the same question — how does a completed sequence first come into existence, before any refinement — and both hand their output to the same max-gap refinement of Algorithm 1, run to $G(x) \leq 0$.

Left-to-right construction. The answer canvas is T slots, all `[MASK]`. Left-to-right construction fills them in index order. At step i the model is given the prompt, the tokens already committed at positions $1, \ldots, i-1$, and `[MASK]` at positions $i, \ldots, T$; it predicts position i, and the highest-probability token there is committed. That is one model call per position, T calls in total, and nothing is revised while construction is in progress: a token written at step i is not reconsidered until all T slots are full. `[MASK]` is removed from the action set during construction, so every step commits a real token rather than leaving the slot empty.

This imitates the *order* of autoregressive decoding without being autoregressive. The backbone is bidirectional and does attend to the positions to the right of i; those positions simply carry no content yet. What they do carry is length — the model can see how many slots remain unwritten — which an autoregressive decoder never sees. Left-to-right construction is therefore a masked model writing in left-to-right order, not a simulation of an autoregressive model.

All-mask construction. All-mask construction does not separate the two phases. The max-gap rule is applied directly to the all-`[MASK]` canvas: the token gap of Equation 3 is evaluated at all T positions, and at a position that has not been written the current token is `[MASK]` itself, which is a legal action. Writing an empty slot and rewriting a filled one are therefore scored on the same scale and compete directly, so the two can interleave. They do: with ModernBERT-Large, 200 of the 1,814 CoQA trajectories and 276 of the 500 PubMedQA trajectories revise an already-written

position before the canvas is complete. Under left-to-right construction that number is zero by definition.

G.6 Results Across Backbones and Datasets

Table 11 extends the initialization comparison to six masked backbones across all three benchmarks. The results reinforce the conclusion from Section 4.2: even with the same prompt, generation budget, banned tokens, and max-gap refinement rule, changing how the initial sequence is constructed can lead to different final equilibrium solutions.

These differences show that initialization and the conditional estimator interact in determining which equilibrium is reached. Each backbone defines its own full-context conditional distributions and therefore its own equilibrium landscape, while the initial construction determines the trajectory followed by Nash decoding within that landscape. Consequently, convergence to a zero-gap solution does not imply that different initializations reach the same equilibrium or achieve the same answer quality. The consistent gains of all-mask initialization for several of the strongest backbones, together with the counterexamples above, indicate that initialization is an important component of equilibrium generation rather than a purely procedural choice.

Table 11: **Left-to-right against all-mask construction, every backbone.** Both rows of a pair share the model, prompt, oracle budget, banned tokens and the max-gap refinement rule; they differ only in how the answer is first written. CoQA uses the GPT-2 paper prompt; PubMedQA and CLAPNQ use the soft-instruction prompt.

		CoQA (1,814)			**PubMedQA** (500)			**CLAPNQ** (300)		
Model	**Size**	F1	R-L	R-Ls	F1	R-L	R-Ls	F1	R-L	R-Ls
Left-to-right construction										
ModernBERT-Base	149M	24.97	28.44	28.39	10.88	12.25	11.46	20.07	22.69	21.38
Ettin-150m	150M	21.74	24.95	24.97	13.28	14.63	13.90	22.54	23.99	23.65
mmBERT-base	308M	42.22	43.18	43.25	4.22	6.11	5.38	9.59	10.01	9.63
RoBERTa-Large	355M	41.84	44.19	44.24	22.23	19.79	20.77	22.86	23.65	23.04
ModernBERT-Large	395M	56.93	57.38	57.41	21.91	19.88	19.99	36.12	33.56	36.07
Ettin-400m	396M	**63.65**	**63.83**	**63.89**	17.41	17.29	17.36	34.68	32.86	34.67
All-mask construction										
ModernBERT-Base	149M	27.00	29.84	29.83	9.16	8.75	8.90	17.28	19.70	19.44
Ettin-150m	150M	21.54	24.10	24.11	16.56	16.62	15.70	23.02	24.56	23.45
mmBERT-base	308M	40.55	41.48	41.61	3.89	5.30	4.59	9.64	9.59	9.63
RoBERTa-Large	355M	41.35	44.02	44.04	24.99	21.17	22.69	25.25	25.59	25.17
ModernBERT-Large	395M	58.12	58.71	58.71	24.28	20.33	21.36	38.04	34.79	37.42
Ettin-400m	396M	61.79	61.98	62.03	22.77	20.05	21.65	34.04	31.61	33.84

H Coordinate Rules on a Shared Diffusion Backbone

Section 4.3 compares Nash decoding with single-token masked-diffusion decoding on one checkpoint. This appendix states that configuration and reports the comparison on all three benchmarks.

H.1 Checkpoint and Coordinate Rules

All three methods use the released `ModernBERT-Large-chat` checkpoint, which keeps the ModernBERT-Large architecture and is supervised fine-tuned on instruction data for masked-diffusion generation. Its identifier is `dllm-hub/ModernBERT-Large-chat-v0.1`. No weights are modified, and a single checkpoint serves every method.

Each method begins from the same all-`[MASK]` canvas and writes one token per step. They differ only in which position is written. *Uniform* order selects a masked position at random; *confidence* order selects the masked position whose predicted token has the highest conditional probability; *Nash gap* selects the position with the largest token gap of Equation 3, evaluated at all T positions rather than only the masked ones.

The two diffusion rules freeze a position once it is written, so each performs exactly T updates and halts. We run them in their single-token setting, with the number of steps and the block length both set to T, so one position is committed per step, and no block structure remains. Sampling is replaced by $\arg\max$ in both methods, so no method draws a random token: the uniform rule is random only in *which* position it fills next. Nash decoding instead recomputes the gap at all T positions after every commitment, may rewrite a position it has already written, and halts only when $G(x) \leq 0$.

H.2 Prompt and Canvas

Every method receives byte-identical input. The prompt is the instruction of Appendix E with one clause appended:

> {`evidence`} **Question:** {`question`}
>
> This question is answered completely with evidence from the passage. Answer using the passage's own words.
>
> **Answer:**

It is presented as raw text, with no chat template, no system message and no added special tokens. The canvas is `[CLS]` prompt `[MASK]`$\times T$ `[SEP]`, where T is the oracle answer-length budget of Appendix E computed in this checkpoint's own tokenizer; its mean is 7.5 tokens on CoQA, 48.3 on PubMedQA and 63.8 on CLAPNQ. We checked that the prompt token ids and the canvas length agree exactly across the three methods on all 2,614 examples, so a method can differ from another only through its coordinate rule.

The longest prompt-plus-canvas in the three datasets is 1,288 tokens, well inside the checkpoint's 8,192-token context, so nothing is truncated. Decoding is greedy in `float32` with TF32 disabled and the SDPA attention implementation. Nash decoding uses tolerance $\varepsilon = 0$, halts at the first repeated state, and is capped at $\max(64, 10T)$ updates.

H.3 Banned Tokens

One action set is used for every dataset and every method. From the checkpoint's vocabulary we remove the four non-mask special ids and eleven further tokens: three pieces of its answer terminator (`[/`, `Answer`, `]`), four chat-template markers (`[`, `SYS`, `Response`, `Question`), and the four newline tokens. That leaves 50,352 actions common to all three methods, plus `[MASK]`, which only Nash decoding may use.

H.4 Results

Table 12 reports the comparison. Nash decoding is ahead of both diffusion rules on all three benchmarks and on all three metrics. Against the stronger of the two rules, which is confidence order everywhere, the paired margin in F1 is $+1.23$ on CoQA (bootstrap CI $[+0.61, +1.82]$, permutation $p = 0.000$), $+0.37$ on PubMedQA ($[+0.09, +0.68]$, $p = 0.008$) and $+0.32$ on CLAPNQ ($[-0.39, +1.03]$, $p = 0.386$).

Table 12: **Nash decoding against single-token masked-diffusion decoding** on the same `ModernBERT-chat` checkpoint. Checkpoint, initialization, prompt, canvas, budget and banned tokens are held fixed, and are the same on all three datasets; only the coordinate-selection rule changes. Best per dataset in **bold**.

	CoQA (1,814)			**PubMedQA** (500)			**CLAPNQ** (300)		
Coordinate rule	F1	R-L	R-Ls	F1	R-L	R-Ls	F1	R-L	R-Ls
Uniform	38.65	40.05	40.08	30.50	23.51	27.14	56.15	50.08	55.00
Confidence	41.85	42.21	42.28	31.74	24.64	28.50	58.84	54.30	58.11
Nash gap	**43.07**	**43.22**	**43.31**	**32.11**	**24.80**	**28.75**	**59.16**	**54.67**	**58.37**

Cost and Convergence. Table 13 gives what the improvement costs and how often the equilibrium criterion is actually met. A diffusion method uses one forward pass per generated token, so its cost is T. Nash decoding rescans all T positions after each commitment, so its cost grows with the product of budget and updates: it is $6.6\times$ the diffusion cost on CoQA, $33\times$ on PubMedQA and $48\times$ on CLAPNQ.

Table 13: **Cost and convergence of the three coordinate rules. Calls** is the mean number of forward passes per example. The diffusion variants always terminate after T commitments.

		Calls	
Dataset	**Mean** T	Diffusion	Nash gap
CoQA (1,814)	7.5	8	53
PubMedQA (500)	48.3	48	1,604
CLAPNQ (300)	63.8	64	3,076

H.5 Empirical cost of Nash decoding

Table 1 reports answer quality under reference-derived token budgets. Figure 6 reports the associated computational cost for ModernBERT-Large with all-mask and left-to-right (L2R) construction, alongside three autoregressive baselines. All five systems were measured on a single B200 under the same numerical settings, with one evaluation process running at a time. CoQA and PubMedQA timings cover their full evaluation sets; CLAPNQ timings cover the shared 60-item subsample specified in the caption. Reported costs include runs that stop at equilibrium or through the cycle and update-cap stopping rules.

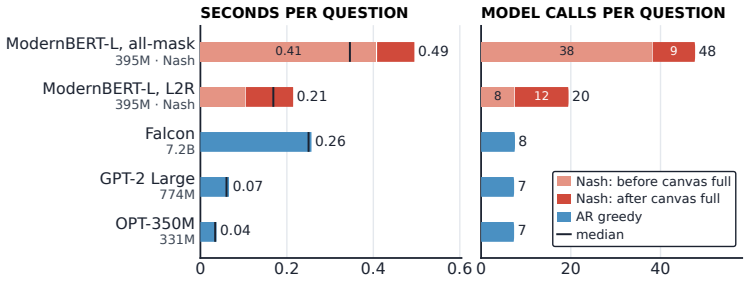
All-mask decoding with ModernBERT-Large takes approximately 0.49, 10.5, and 17.4 s per question on the timed CoQA, PubMedQA, and CLAPNQ sets, respectively, requiring approximately 48, 1,520, and 2,710 model calls. Falcon, the 7.2B-parameter autoregressive baseline, takes approximately 0.26, 0.8, and 1.1 s on the same sets. These values follow the rounding used in Figure 6.

The larger overhead on the longer-answer tasks is consistent with the work required to recompute token gaps. For a canvas of T positions, a full gap scan can require up to T conditional evaluations. Repeating this scan across K token replacements therefore requires $O(T(K+1))$ conditional evaluations, including the final scan. Actual wall-clock cost also depends on prompt length, the number of replacements, the backbone, and how the evaluations are batched.

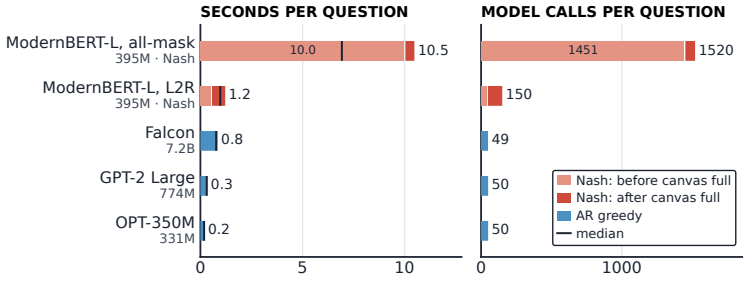
L2R construction reduces this cost. It requires approximately 20, 150, and 249 calls on the three timed sets, compared with 48, 1,520, and 2,710 for all-mask decoding: reductions of approximately $2.4\times$, $10.1\times$, and $10.9\times$, respectively. On the full evaluation sets, the corresponding ModernBERT-Large F1 scores are 56.93 versus 58.12 on CoQA, 21.91 versus 24.28 on PubMedQA, and 36.12 versus 38.04 on CLAPNQ. Thus, L2R construction trades approximately 1.2, 2.4, and 1.9 F1 points for lower computational cost. The CLAPNQ quality comparison uses all 300 questions, whereas its displayed timing comparison uses the 60-item subsample.

The two constructions also distribute computation differently. L2R fills the canvas in T sequential predictions before refinement begins, so subsequent refinement accounts for most of its model calls in these measurements. All-mask decoding repeatedly evaluates competing positions while the canvas is incomplete; most of its calls occur before the last mask is filled, and this interval can already include revisions.

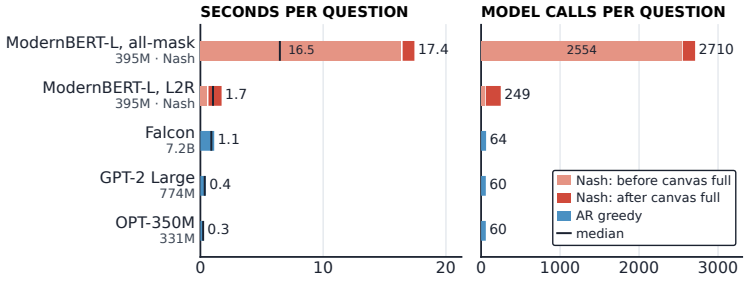
Model-call counts should be interpreted alongside wall-clock times because a call represents different amounts of work across the two model families. After its initial prompt evaluation, an autoregressive decoder processes each new token using a cached prefix, whereas the masked implementation re-encodes the prompt-and-canvas sequence for its conditional evaluations. The timings therefore characterize these backbones and this implementation under the stated hardware and numerical settings. They do not establish an implementation-independent cost of finding an equilibrium.



(a) CoQA



(b) PubMedQA



(c) CLAPNQ

Figure 6: Wall-clock seconds and model calls per question for the same five systems on each dataset, measured on a single B200 in `float32` with TF32 disabled and one evaluation process running at a time. The Nash bars are split at the update that first fills the last `[MASK]`: the light segment includes all computation through that update, and the dark segment includes subsequent refinement. For all-mask decoding, the light segment can include revisions of previously filled positions. Horizontal scales differ between panels. CoQA and PubMedQA use their full evaluation sets. CLAPNQ uses a fixed 60-item subsample, consisting of every fifth item and shared by all five systems. Its mean reference answer length is 49.0 words, compared with 51.7 for the full set; mean L2R cost is 249 calls, compared with 248 on all 300 items.

I Use of Generative AI

Generative AI tools were used to assist with implementing the experiments, developing and refining code for data visualization and figure generation, and editing the manuscript for grammar, clarity, and readability. All AI-assisted outputs were reviewed, tested where applicable, and revised by the authors, who take full responsibility for the final manuscript, experimental results, and conclusions.

J Code and Implementation Details

To support reproducibility, we provide the complete implementation as supplementary material with this submission, including the code and scripts required to reproduce the experiments, tables, and figures reported in the paper. Following the double-blind review period, we will release the full codebase in a public GitHub repository together with the corresponding experimental configurations.