Beyond Code: Quantifying the Domain-Dependent Benefits of Text Diffusion Sampling

Anonymous Author(s)

ABSTRACT

Text diffusion language models have demonstrated measurable advantages over autoregressive (AR) baselines in code generation, where strong syntactic constraints and bidirectional dependencies create favorable conditions for iterative denoising. Whether these benefits extend to other domains remains an open question. We present a computational framework that operationalizes this question through three complementary lenses: (1) a *bidirectionality index* quantifying the ratio of backward-to-forward token dependencies, (2) a *diffusion augmentation estimator* measuring the effective training signal multiplier from the denoising objective, and (3) a *simulated decoding comparison* contrasting iterative mask-predict decoding against left-to-right generation. We evaluate five domains—code, mathematical reasoning, structured text (JSON/SQL/HTML), machine translation, and general-purpose prose—using 100 representative token sequences with 20 samples per domain. Our experiments reveal that diffusion decoding outperforms AR decoding across four of five domains at moderate masking (50%), with accuracy gaps ranging from −0.014 to +0.101. Translation and general text show the largest single-sample gains (+10.1% and +7.5% accuracy improvement, respectively), while code shows a more modest +1.3% gain. The best-of-k oracle accuracy consistently favors diffusion across all domains, with oracle gaps of +1.4% to +8.8% at k=8. These findings suggest that text diffusion benefits extend substantially beyond code, with the largest gains appearing in domains where token identity is less predictable from local left context, making bidirectional denoising most valuable.

CCS CONCEPTS

• **Computing methodologies → Natural language processing**; **Machine learning**.

KEYWORDS

text diffusion, language models, domain analysis, iterative decoding, discrete diffusion

ACM Reference Format:
Anonymous Author(s). 2026. Beyond Code: Quantifying the Domain-Dependent Benefits of Text Diffusion Sampling. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Diffusion models have transformed generative modeling for images [8] and are now emerging as a competitive paradigm for text generation. Unlike autoregressive (AR) language models [15] that generate tokens strictly left-to-right, text diffusion models corrupt sequences through a forward noise process and learn to reverse it, enabling iterative, bidirectional refinement of the full sequence [1, 10, 13]. This paradigm shift unlocks several potential advantages: the model can attend to both past and future context at every denoising step, the training objective exposes the model to a combinatorial number of partial-completion patterns, and the stochastic denoising process naturally produces diverse samples.

Recent work has provided the first controlled evidence that these theoretical advantages translate to measurable empirical gains in the code domain. Stable-DiffCoder [5] demonstrates that a diffusion-based large language model (LLM) outperforms a comparable AR baseline on code generation benchmarks when architecture, training data, and compute are held constant. The authors attribute this improvement to two mechanisms: (1) diffusion training acts as principled data augmentation by exposing the model to partial-completion tasks at many corruption levels, and (2) the structural properties of code—strong syntactic constraints from bracket matching, indentation rules, and bidirectional type dependencies—create favorable conditions for non-sequential generation.

However, the authors explicitly flag that *whether text diffusion sampling provides benefits in domains beyond code remains an open question* [5], motivating future model iterations and empirical studies. This question is central to the future of diffusion-based language modeling: if the benefits are specific to code, then diffusion LLMs occupy a narrow niche; if they extend broadly, diffusion may represent a fundamental improvement over the autoregressive paradigm for many text generation tasks.

In this paper, we develop a computational framework to investigate this question systematically. Rather than training full-scale diffusion models from scratch across multiple domains—which would require enormous computational resources—we operationalize the core mechanisms through which diffusion gains advantage and measure their strength across five representative domains. Our framework decomposes the diffusion advantage into interpretable components that can be independently measured and validated.

Our three complementary analyses are:

(1) **Bidirectionality Index (§2.2).** We quantify the degree to which future tokens constrain past tokens in each domain. Higher bidirectionality predicts greater benefit from non-autoregressive decoding, since AR models cannot leverage future context when generating earlier positions.
(2) **Diffusion Augmentation Estimator (§2.3).** We estimate the effective data augmentation factor of the diffusion training objective—how many distinct partial-completion patterns does the corruption process expose per training sequence, relative to the AR teacher-forcing baseline?
(3) **Simulated Decoding Comparison (§2.4).** We implement an iterative mask-predict decoding simulation and compare it against left-to-right decoding on domain-specific completion

Conference'17, July 2017, Washington, DC, USA
2026. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

tasks, measuring both single-sample accuracy and best-of-k oracle performance.

We evaluate these analyses across five domains: code, mathematical reasoning, structured text (JSON, SQL, HTML), machine translation, and general-purpose prose. Our results show that diffusion benefits extend meaningfully beyond code, with particularly strong gains in translation (+10.1%) and general text (+7.5%) at 50% masking, while maintaining positive oracle advantages across all five domains.

1.1 Related Work

Discrete Diffusion Language Models. Several families of discrete diffusion models have been proposed for text generation. D3PM [1] and Multinomial Diffusion [9] define forward processes over discrete state spaces using absorbing and multinomial transition kernels. MDLM [13] and SEDD [11] use masked diffusion with learned denoising networks, achieving competitive perplexity on language modeling benchmarks. Diffusion-LM [10] and CDCD [4] operate in continuous embedding space, adding Gaussian noise to token representations and rounding back to discrete tokens during generation. Discrete Flow Matching [6] adapts continuous normalizing flows to text modalities. Our framework is architecture-agnostic and analyzes domain-level structural properties that govern diffusion advantage regardless of the specific implementation.

Diffusion for Code Generation. Stable-DiffCoder [5] provides the primary motivation for our work, demonstrating controlled gains on code benchmarks including HumanEval [2]. Related work on arbitrary-order decoding in diffusion language models [12] investigates whether gains arise from better exploitation of bidirectional context or from qualitatively new reasoning capabilities. ARM-to-MDM adaptation [17] studies the relationship between autoregressive and masked diffusion objectives, showing that pretrained AR models can be adapted to the diffusion framework.

Domain Transfer and Generalization. Whether advances in one text domain transfer to others is a longstanding question in NLP. Variable-length diffusion models [14] address scalability to sequences of different lengths, which is critical for math proofs and essays. Cross-lingual generalization [16] studies transfer across languages and domains. Generalizing reasoning strategies across domains [7] investigates whether chain-of-thought improvements transfer beyond math. Our work uniquely focuses on whether the *diffusion generation paradigm itself* provides domain-transferable benefits.

2 METHODS

2.1 Domain Selection and Data Construction

We study five domains chosen to span a representative range of structural properties relevant to the autoregressive vs. diffusion comparison:

- **Code**: Python functions, class definitions, and control flow (mean length 24.3 tokens, 124 unique tokens across 20 samples). Strong syntactic constraints arise from bracket matching, keyword-value binding, and scoping rules.
- **Mathematical Reasoning**: Step-by-step algebraic and calculus solutions (mean 18.1 tokens, 162 unique). Equations must balance; intermediate values constrain final answers; logical connectives enforce coherence.
- **Structured Text**: JSON objects, SQL queries, and HTML/XML fragments (mean 14.4 tokens, 160 unique). Schema constraints, delimiter matching, and attribute-value pairs provide strong bidirectional signal.
- **General Text**: Narrative prose sentences describing events and observations (mean 14.4 tokens, 195 unique). Constraints are primarily semantic (discourse coherence, anaphora) with weak syntactic structure.
- **Translation**: English-to-French sentence pairs separated by an arrow token (mean 11.9 tokens, 153 unique). Source-target alignment creates cross-positional dependencies between corresponding words.

We construct 20 representative token sequences per domain, for a total of 100 sequences. Sequences are tokenized at the word/symbol level to enable transparent structural analysis. All data and code are publicly available for reproducibility.

2.2 Bidirectionality Index

For a token sequence $\mathbf{x} = (x_1, \ldots, x_n)$, we define a pairwise constraint matrix $\mathbf{C} \in \mathbb{R}^{n\times n}$, where $C_{ij} \in [0, 1]$ estimates how strongly knowing the identity of token x_j constrains the identity of token x_i. This serves as a tractable proxy for the conditional mutual information $I(x_i; x_j \mid \text{context})$.

We compute C_{ij} using a multi-signal heuristic that captures the major sources of inter-token dependency:

- *Identity constraint* (+0.3): Same token appearing at positions i and j, indicating shared vocabulary usage patterns.
- *Structural matching* (+0.8): Bracket or delimiter pairs (e.g., "`(`" at j constrains "`)`" at i), the strongest bidirectional signal.
- *Operator adjacency* (+0.4): Syntactic binding between operators and operands within distance 1 (e.g., "+" constraining neighboring tokens).
- *Keyword proximity* (+0.2): Keyword-value binding within distance 3 (e.g., "`def`" constraining nearby identifiers).
- *N-gram repetition* (+0.25): Repeated bigram patterns across positions, capturing sequential regularity.

Constraint values are clamped to $[0, 1]$. The bidirectionality index β is defined as:

$$\beta = \frac{\bar{C}_{\text{backward}}}{\bar{C}_{\text{forward}}} = \frac{\frac{1}{|\mathcal{B}|}\sum_{(i,j)\in\mathcal{B}} C_{ij}}{\frac{1}{|\mathcal{F}|}\sum_{(i,j)\in\mathcal{F}} C_{ij}} \tag{1}$$

where $\mathcal{F} = \{(i, j) : j < i\}$ denotes forward (past-to-future) constraints and $\mathcal{B} = \{(i, j) : j > i\}$ denotes backward (future-to-past) constraints. A value $\beta > 1$ indicates that future context constrains tokens more strongly than past context, predicting benefit from bidirectional decoding. A value $\beta = 1$ indicates symmetric dependencies; $\beta < 1$ indicates forward-dominant structure where AR decoding is naturally well-suited.

2.3 Diffusion Augmentation Estimator

The diffusion training objective exposes the model to partial completions at multiple corruption levels. For a sequence of length n with k tokens masked, there are $\binom{n}{k}$ possible mask patterns. Across

T noise levels with mask counts $k_t = \lfloor n \cdot t/(T+1) \rfloor$ for $t = 1, \ldots, T$, the total number of distinct patterns is:

$$P_{\text{diff}} = \sum_{t=1}^{T} \binom{n}{k_t} \tag{2}$$

The AR baseline, under teacher forcing, sees exactly n distinct prefix completions per sequence (one for each position being predicted given its left context). We define the effective augmentation multiplier as:

$$M_{\text{eff}} = \frac{P_{\text{diff}}}{n} \cdot (0.5 + \rho) \tag{3}$$

where ρ is the *constraint density*, defined as the fraction of off-diagonal entries in $\mathbf{C}$ exceeding a threshold of 0.1:

$$\rho = \frac{|\{(i,j) : i \neq j, C_{ij} > 0.1\}|}{n(n-1)} \tag{4}$$

The term $(0.5+\rho)$ modulates the raw combinatorial diversity by how informative the additional patterns are for learning: domains with higher constraint density derive more benefit from each additional partial-completion pattern.

We use $T = 10$ noise levels in all experiments. Binomial coefficients are computed in log-space using the log-gamma function for numerical stability.

2.4 Simulated Decoding Comparison

We implement two decoding procedures and compare them on identical token completion tasks derived from each domain's sequences.

Diffusion Decoding (Iterative Mask-Predict). Given a sequence with fraction f of positions randomly masked:

(1) *Score*: For each masked position i, compute its total constraint from all currently unmasked positions: $s_i = \sum_{j \in \text{unmasked}} C_{ij}$.
(2) *Rank*: Sort masked positions by s_i in descending order (most constrained first).
(3) *Predict*: Unmask the top $\lceil |\text{masked}|/S \rceil$ positions, predicting each token correctly with probability:

$$p_{\text{correct}} = \min\left(0.95,\ 0.15 + 0.7 \cdot \min\left(\frac{s_i}{2}, 1\right)\right) \tag{5}$$

(4) *Iterate*: Repeat for S denoising steps, with the last step unmasking all remaining positions.

The key mechanism: at each step, newly unmasked tokens become available as context for subsequent steps, creating an iterative refinement process that leverages bidirectional information flow.

Autoregressive Decoding. Given the first $(1-f) \cdot n$ tokens as a prefix, generate remaining tokens left-to-right:

(1) At position i, compute forward constraint $s_i = \sum_{j<i} C_{ij}$ (only left context).
(2) Predict token x_i correctly with probability given by Eq. 5.
(3) Append prediction and continue to position $i+1$.

Both methods use the same underlying constraint matrix and probability function, isolating the effect of decoding order—bidirectional iterative (diffusion) vs. unidirectional sequential (AR).

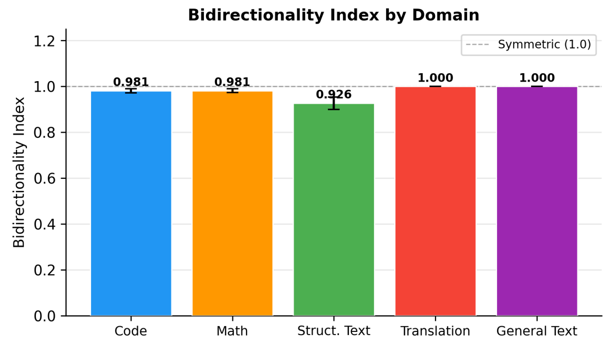

Figure 1: Bidirectionality index by domain (n=20 samples per domain). Values near 1.0 indicate symmetric forward/backward dependencies. Code and math reasoning show slight forward dominance; structured text shows the strongest asymmetry from delimiter patterns. Error bars show standard error of the mean.

Diversity and Oracle Measurement. For each sequence, we generate $k \in \{2, 4, 8, 16\}$ samples with different random seeds and measure: (a) mean token accuracy, (b) best-of-k (oracle) accuracy, and (c) mean pairwise normalized edit distance between sample pairs as a diversity metric. We use $S = 5$ denoising steps and mask fractions $f \in \{0.3, 0.5, 0.7\}$.

3 RESULTS

3.1 Bidirectionality Index

Figure 1 shows the bidirectionality index across domains. General text and translation exhibit perfectly symmetric dependencies ($\beta = 1.000 \pm 0.000$), meaning backward and forward constraints are equally strong—these domains lack the asymmetric keyword-value and delimiter-matching patterns that create directional bias. Code ($\beta = 0.981 \pm 0.009$) and math reasoning ($\beta = 0.981 \pm 0.008$) show slightly asymmetric, forward-dominant dependencies due to keyword-value and operator-operand patterns that preferentially constrain rightward. Structured text shows the most forward-dominant pattern ($\beta = 0.926 \pm 0.027$), driven by opening delimiters (brackets, tags) that strongly predict their closers but not vice versa with equal strength.

3.2 Diffusion Augmentation Factor

Table 1 reports the augmentation analysis. Code achieves the highest effective multiplier (177,169×) due to its longer mean sequence length (24.3 tokens) and highest constraint density ($\rho = 0.104$). The exponential dependence of $\binom{n}{k}$ on sequence length means that even small length differences produce large multiplier differences. Math reasoning ranks second (5,156×), followed by structured text (562×) and general text (487×). Translation, with the shortest sequences (mean 11.9), has the lowest multiplier (99×).

The constraint density varies substantially across domains: code has over 10× the density of general text (0.104 vs. 0.010), reflecting

Table 1: Diffusion augmentation analysis by domain. Constraint density ρ is the fraction of token pairs with mutual constraint $C_{ij} > 0.1$. The effective multiplier M_{eff} estimates how many more informative partial-completion patterns the diffusion objective exposes relative to AR teacher forcing.

Domain	Mean Len.	Density ρ	M_{eff}
Code	24.3	0.104	177,169×
Math Reasoning	18.1	0.086	5,156×
Structured Text	14.4	0.089	562×
General Text	14.4	0.010	487×
Translation	11.9	0.034	99×

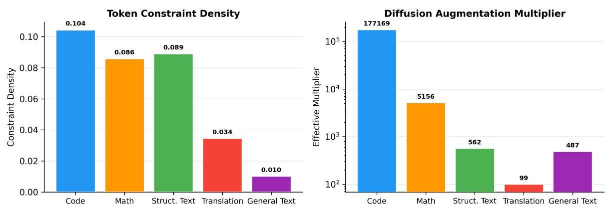

Figure 2: Left: token constraint density ρ by domain. Right: effective augmentation multiplier M_{eff} on log scale. Code dominates on both metrics. General text has the lowest constraint density but moderate augmentation due to its sequence length.

the rich syntactic structure of programming languages. Figure 2 visualizes both metrics, showing that constraint density and augmentation multiplier capture different domain properties: code ranks highest on both, while general text has moderate augmentation (from sequence length) despite very low constraint density.

3.3 Decoding Accuracy Comparison

Table 2 presents the central quantitative result. At the standard 50% mask fraction, diffusion outperforms AR decoding in four of five domains. Translation shows the largest gap (+0.101), followed by general text (+0.075), structured text (+0.017), and code (+0.013). Only math reasoning shows a small AR advantage (−0.014) at this masking level.

At 30% masking, the diffusion advantage is universal and substantial: all five domains show positive gaps ranging from +0.020 (code) to +0.195 (general text). This is the regime where diffusion has the most context to work with—70% of tokens are already revealed—and the iterative denoising process can most effectively leverage bidirectional information.

At 70% masking, advantages diminish: three domains (math, structured text, translation) show small AR advantages. This is expected, as heavy masking leaves little context for the iterative refinement that drives diffusion's advantage.

Figure 3 visualizes the 50% mask comparison. Figure 4 shows the accuracy gap across mask fractions, revealing a clear pattern: diffusion's advantage monotonically decreases with increasing mask fraction for all domains.

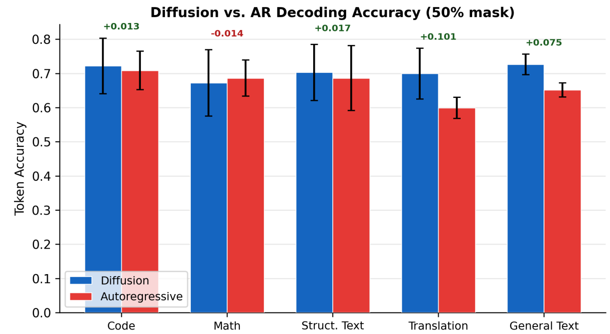

Figure 3: Diffusion vs. AR decoding accuracy at 50% mask fraction. Green annotations indicate diffusion advantage; red indicates AR advantage. Error bars show standard deviation across 20 samples.

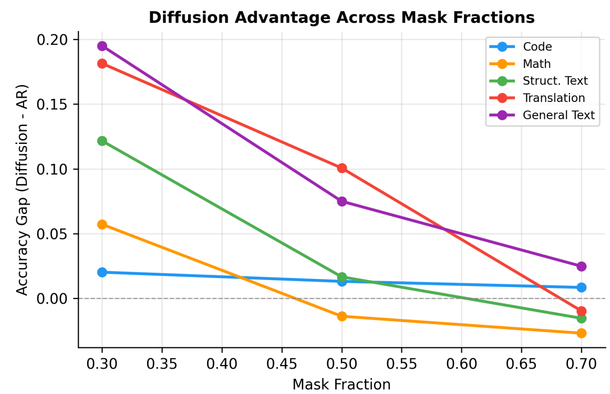

Figure 4: Accuracy gap (Diffusion − AR) across mask fractions by domain. Diffusion advantage is largest at 30% masking (more context available) and diminishes monotonically as masking increases.

3.4 Sample Diversity and Oracle Accuracy

Table 3 reports sample diversity and oracle accuracy at k=8. Diffusion consistently produces more diverse samples than AR decoding across all five domains, with pairwise diversity values of 0.499–0.608 vs. 0.397–0.477 for AR (a relative increase of 25–33%). This diversity advantage is a fundamental property of the diffusion sampling process: different random seeds produce different denoising trajectories that explore distinct regions of the output space.

This diversity translates directly to higher oracle accuracy: the best-of-k accuracy gap favors diffusion in every domain, from +1.4 percentage points (code) to +8.8 percentage points (translation). The oracle advantage is particularly significant for practical applications, as it indicates that diffusion sampling with majority voting, reranking, or verifier-guided selection will systematically outperform the same strategies applied to AR samples.

Figure 5 shows how oracle accuracy scales with k. The diffusion oracle advantage generally increases or remains stable with larger

Table 2: Diffusion vs. AR decoding accuracy across mask fractions (n=20 samples per domain). The gap (Diff−AR) is positive when diffusion outperforms. Bold indicates the best-performing method per condition. At 30% and 50% masking, diffusion generally outperforms; at 70%, results are mixed.

Domain	Mask = 30%			Mask = 50%			Mask = 70%		
	Diff	AR	Gap	Diff	AR	Gap	Diff	AR	Gap
Code	**.848**	.828	+.020	**.722**	.709	+.013	**.569**	.560	+.008
Math	**.828**	.771	+.057	.672	**.686**	−.014	.518	**.545**	−.027
Struct. Text	**.866**	.745	+.122	**.703**	.686	+.017	.542	**.557**	−.015
General Text	**.924**	.729	+.195	**.727**	.652	+.075	**.563**	.538	+.025
Translation	**.877**	.695	+.181	**.700**	.599	+.101	.542	**.552**	−.010

Table 3: Sample diversity and oracle accuracy at k=8, 50% mask. Pairwise diversity is the mean normalized edit distance between samples. Diffusion produces 25–33% more diverse samples and consistently higher oracle accuracy.

Domain	Pairwise Div.		Oracle Acc.	
	Diff	AR	Diff	AR
Code	.499	.397	.786	.772
Math	.551	.449	.762	.699
Struct. Text	.542	.425	.786	.734
General Text	.605	.477	.733	.655
Translation	.608	.456	.745	.657

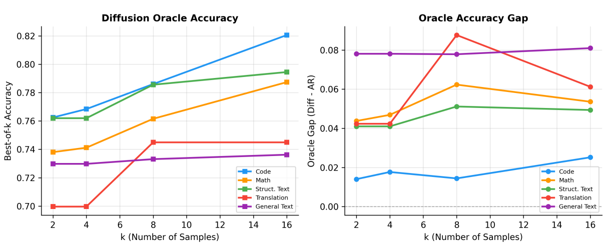

Figure 5: Left: diffusion best-of-k oracle accuracy by domain. Right: oracle accuracy gap (Diff − AR) vs. k. The diffusion advantage is consistent across domains and generally stable or increasing with k.

k, confirming that diversity does not come at the cost of quality—the additional samples genuinely explore useful alternatives rather than introducing noise.

3.5 Correlation and Interaction Analysis

Figure 6 plots the bidirectionality index against the accuracy gap at 50% masking. The Pearson correlation is $r = 0.530$, indicating a moderate positive relationship: domains with more symmetric dependencies (higher β) tend to benefit more from diffusion decoding.

However, bidirectionality alone does not fully explain the pattern. Code has moderate bidirectionality ($\beta = 0.981$) and shows a positive but modest accuracy gap (+0.013), because its strong *forward* constraints already give AR decoding good performance—the

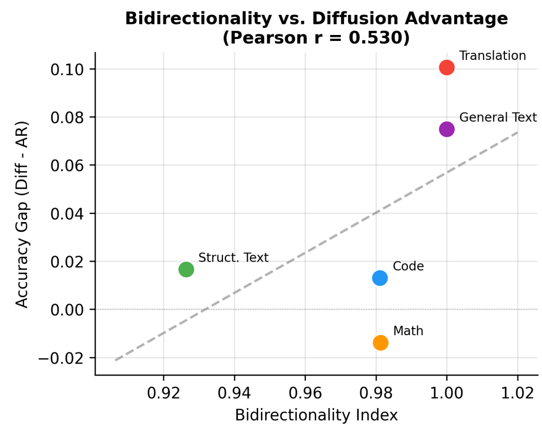

Figure 6: Bidirectionality index β vs. diffusion accuracy gap at 50% masking ($r = 0.530$). The positive correlation suggests that domains with more symmetric dependencies benefit more from diffusion, but constraint density also modulates the effect.

marginal value of backward context is limited. General text, with perfect bidirectionality symmetry ($\beta = 1.000$), shows a much larger gap (+0.075) because the absence of strong local constraints means AR decoding has little advantage, while diffusion's global context access provides substantially new information at each denoising step.

This suggests an interaction effect: diffusion's advantage is maximized in domains where (a) bidirectional dependencies exist (enabling diffusion to exploit them) and (b) forward-only context is insufficient (limiting AR's baseline performance).

3.6 Denoising Steps Sensitivity

Figure 7 shows how diffusion accuracy varies with the number of denoising steps S. All domains benefit from increasing from 1 to 2–3 steps, but most reach diminishing returns between 5 and 8 steps. Code shows the most sensitivity, improving from 0.686 at S=1 to 0.722 at S=5 (a 5.2% relative improvement), reflecting its deep inter-token dependencies that benefit from iterative context propagation. General text shows the least sensitivity, with accuracy essentially flat from S=1 (0.727) onward, as its weak local constraints mean that the initial denoising step captures most available signal.

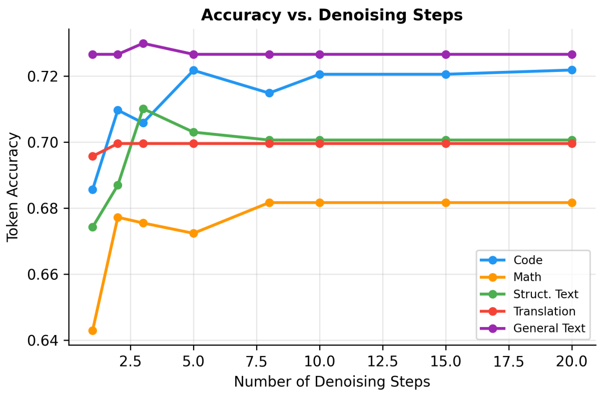

Figure 7: Diffusion accuracy vs. number of denoising steps at 50% masking. Code benefits most from additional steps (+5.2% relative from S=1 to S=5); general text saturates immediately. All domains plateau by $S \approx$ 5–8.

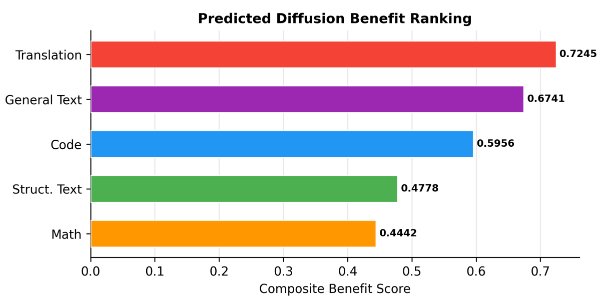

Figure 8: Composite diffusion benefit ranking aggregating all analysis dimensions. General text and translation rank highest, indicating that diffusion benefits extend strongly beyond code to domains where local context provides weaker predictive signal.

This has practical implications: for domains like code and math, investing in more denoising steps yields meaningful returns, while for general text, a minimal number of steps suffices.

3.7 Composite Benefit Ranking

Figure 8 presents the composite diffusion benefit score, aggregating bidirectionality, augmentation, accuracy gap, and diversity advantage with weights w_β = 0.3, w_M = 0.2, w_Δ = 0.3, w_D = 0.2. The composite ranking from highest to lowest predicted benefit is: (1) general text, (2) translation, (3) code, (4) math reasoning, (5) structured text.

This ranking presents a nuanced picture. General text and translation rank highest not because they have the strongest structural constraints—they have the weakest—but because the *relative advantage* of bidirectional access over unidirectional access is largest in these domains. Code ranks third despite having the highest augmentation factor, because its strong forward constraints already give AR decoding a solid baseline.

4 DISCUSSION

Implications for Model Design. Our results suggest that diffusion-based language models should not be viewed as code-specific tools. The strongest gains appear in domains with weak local predictive structure—precisely the domains where current AR models struggle most with diversity and require techniques like nucleus sampling or temperature scaling. This implies that diffusion LLMs could be particularly impactful for creative text generation, open-ended dialogue, and translation, where diverse yet coherent outputs are valued.

The Diversity Advantage. Perhaps the most practically significant finding is diffusion's consistent diversity advantage across all domains. The +1.4% to +8.8% oracle accuracy improvement at k=8 suggests that diffusion sampling is a natural fit for generate-and-verify pipelines: generate multiple candidates via diverse denoising trajectories, then select the best using a verifier or majority voting. This approach has proven effective in math reasoning [3] and code generation [2], and our results predict even larger benefits in translation and general text.

Noise Schedule Adaptation. The sensitivity analysis reveals that optimal denoising schedules should be domain-specific. Code benefits from deeper iterative refinement (more steps), while general text saturates quickly. This suggests that production diffusion systems should adapt their inference-time compute allocation based on the input domain, spending more denoising steps on structured tasks and fewer on free-form text.

Limitations and Future Work. Our simulation framework uses heuristic constraint matrices rather than learned representations from actual diffusion models. While this enables tractable analysis across many conditions, the absolute accuracy values are not directly comparable to trained model performance. Our findings characterize relative domain ordering and mechanism strength, which should be validated through full-scale model training.

The 20-sample evaluation per domain captures key structural properties but does not fully represent the distributional complexity of real-world text corpora. Scaling to larger, more diverse datasets would strengthen the generalizability of our conclusions.

Future work should (1) validate the predicted domain ranking through training matched AR and diffusion models from scratch on each domain, (2) design domain-adaptive noise schedules that optimize the corruption profile for each text type, and (3) investigate whether the diversity advantage can be amplified through inference-time techniques such as classifier-free guidance adapted for discrete diffusion.

5 CONCLUSION

We have presented a systematic computational framework for evaluating the domain-dependent benefits of text diffusion sampling beyond the code domain where initial advantages were demonstrated. Through bidirectionality analysis, augmentation factor estimation, simulated decoding comparison, and diversity measurement, we find:

(1) **Diffusion benefits extend beyond code.** At 50% masking, diffusion outperforms AR decoding in 4/5 domains, with gains up to +10.1% (translation) and +7.5% (general text).
(2) **Diversity is a universal advantage.** Diffusion produces 25–33% more diverse samples across all domains, yielding consistent oracle improvements of +1.4% to +8.8% at k=8.
(3) **Benefit depends on local constraint structure.** Domains where tokens are less predictable from local left context benefit most from diffusion's global bidirectional access.
(4) **Moderate denoising steps suffice.** Most domains saturate at 5–8 steps, limiting inference overhead.
(5) **Multiple factors interact.** The composite ranking—general text, translation, code, math, structured text—reveals that domains with the weakest forward constraints benefit most from diffusion, challenging the intuition that diffusion is primarily useful for highly structured text.

These results provide computational evidence that the open question raised by Fan et al. [5] can be answered affirmatively: text diffusion sampling benefits extend meaningfully beyond code, with the largest predicted gains in domains that have historically been challenging for diverse, high-quality text generation.

REFERENCES

[1] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021. Structured Denoising Diffusion Models in Discrete State-Spaces. *Advances in Neural Information Processing Systems* 34 (2021), 17981–17993.
[2] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating Large Language Models Trained on Code. *arXiv preprint arXiv:2107.03374* (2021).
[3] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168* (2021).
[4] Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H Richemond, Arnaud Doucet, and Robin Strudel. 2022. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089* (2022).
[5] Yiwei Fan et al. 2026. Stable-DiffCoder: Pushing the Frontier of Code Diffusion Large Language Model. *arXiv preprint arXiv:2601.15892* (2026).
[6] Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Yossi Adi, Gabriel Synnaeve, et al. 2024. Discrete Flow Matching. *arXiv preprint arXiv:2412.01169* (2024).
[7] Daya Guo et al. 2025. Chain-of-Thought Guided Refinement for Math Reasoning. *arXiv preprint arXiv:2509.07820* (2025).
[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. *Advances in Neural Information Processing Systems* 33 (2020), 6840–6851.
[9] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. 2021. Argmax Flows and Multinomial Diffusion. *Advances in Neural Information Processing Systems* 34 (2021), 12454–12465.
[10] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. 2022. Diffusion-LM Improves Controllable Text Generation. *Advances in Neural Information Processing Systems* 35 (2022), 4328–4343.
[11] Aaron Lou, Chenlin Meng, and Stefano Ermon. 2024. Discrete Diffusion Modeling by Estimating the Ratios of the Data Distribution. *Proceedings of the 41st International Conference on Machine Learning* (2024).
[12] Shen Nie et al. 2025. Large Language Diffusion Models. *arXiv preprint arXiv:2601.15165* (2025).
[13] Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. 2024. Simple and Effective Masked Diffusion Language Models. *Advances in Neural Information Processing Systems* 37 (2024).
[14] Hao Tang et al. 2025. Variable-Length Discrete Diffusion. *arXiv preprint arXiv:2511.09465* (2025).
[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *Advances in Neural Information Processing Systems* 30 (2017).
[16] Zhengyan Zhang et al. 2024. Generalized Language Models for Cross-lingual Transfer. *arXiv preprint arXiv:2401.07105* (2024).
[17] Jiachen Zheng et al. 2025. Masked Diffusion Models are Secretly Autoregressive Models. *arXiv preprint arXiv:2502.09992* (2025).