Anchor-Token Masking Produces an Out-of-Distribution Generalization Advantage in Retrieval-Grounded Masked-Diffusion LMs

Thomas Garren[1] [1]SOPHIA XT LLC — `thomas@sophiaxt.com` — `sophiaxt.com` *Independent research* · *Technical report* · *2026-05-09*

Abstract

Masked-diffusion language models (MDLMs) admit a class of training objectives that asymmetrically weight prefix and target positions, paralleling prefix-LM training in the autoregressive setting. We empirically compare three masking strategies on identical (Q, A) training data: *anchor-token masking* (Q never masked, A always subject to schedule), *random-position masking* (uniform over Q ∪ A), and a *reverse-asymmetric* control (Q always subject to schedule, A never masked). Using rank-16 LoRA adapters trained on 144 retrieval-grounded examples for 300 steps over a 1.3B-parameter undertrained MDLM (Cassandra T1, epoch-5 cross-entropy 2.26), we measure the corpus-vocabulary overlap of generated text around forced-anchor positions on the LoRA's training corpora and on **four held-out corpora** spanning astronomy, medical/anatomy, legal/contracts, and cooking/cuisine domains (10.5–15.6% vocabulary overlap with training).

Anchor-token masking produces a pooled **1.67× generalization advantage** over random-position masking across N=48 held-out queries (95% bootstrap CI [1.51×, 1.85×], P(anchor > random) = 1.000). The effect replicates with P = 1.000 on three of four held-out domains individually (astronomy 2.05×, medical 1.72×, legal 2.20×) and weakly with P = 0.994 on the fourth (cooking 1.19×, attributable to higher generic-vocabulary overlap with training corpora). In-domain, the anchor advantage over random is 1.26× [1.18×, 1.34×]. The held-out and in-domain confidence intervals do not overlap, indicating that anchor-token masking produces a *generalization-favoring* effect, not just an in-distribution fit advantage.

Mechanism probe 1 — direction: the reverse-asymmetric LoRA performs statistically significantly *worse* than the random-position LoRA on held-out (diff = −0.038, 95% CI [−0.068, −0.008], P(reverse > random) = 0.006), giving a clean three-arm ranking: anchor > random > reverse ≫ base. This rules out the "asymmetric optimization landscape alone" hypothesis: the *direction* of asymmetry is load-bearing, not just its presence.

Mechanism probe 2 — attention: instrumenting attention weights during forced-anchor generation, we find the anchor-LoRA shows the highest attention to anchor positions (anchor_share = 0.272 vs random 0.259 vs base 0.256), but the magnitude is small relative to the behavioral effect (~10% of the behavioral gap). The simplest "model learns to look at anchors" mechanism is therefore not the primary driver.

Mechanism probe 3 — weight allocation: computing Frobenius norms of effective LoRA delta-weight matrices per projection type, we find anchor-LoRA is the only configuration where V/O (value/output) capacity exceeds Q/K (attention pattern) capacity (ratio 1.15 vs random 0.79 vs reverse 0.94). The anchor-token-masked training objective frees LoRA capacity from Q-projection updates (Q is always visible during training) and reallocates it to V/O — which control what gets extracted from attended positions, not where the model attends.

Mechanism probe 4 — V/O ablation (causal test): we train two ablation LoRAs at half the trainable parameters (2.98M vs 7M). The V/O-only LoRA (q_proj/k_proj frozen) achieves **0.416 held-out overlap — exceeding the full LoRA at 0.337**, indicating Q/K updates were actively counterproductive. The Q/K-only LoRA (v_proj/o_proj frozen) achieves only 0.136 — *below* the random-position-mask full LoRA control at 0.214 — establishing that Q/K alone cannot carry the effect. V/O is therefore both necessary and sufficient for the inductive bias. This is direct causal confirmation of the capacity-reallocation hypothesis.

Sample-efficiency sweep: across six (data-size × step-count) configurations, the anchor advantage peaks at 1.97-2.19× at low budgets and attenuates monotonically with step count, collapsing to 1.07× at 1000 steps on 144 examples. The technique is best characterized as a low-shot fine-tuning advantage, with optimal regime at ~100-300 steps on 50-150 examples.

Qualitative analysis shows a clean behavioral distinction: anchor-trained extends the locked anchor's topical content (*"the planet features a radius," "transit period"* on astronomy); random-trained falls back to training-corpus tokens (*"Cassandra T1, a diffusion language model by SOPHIA XT"*); reverse-trained generates question-template fragments (*"Tell me about"*) — each consistent with its training-objective inductive bias.

We synthesize a causally-supported mechanism account: anchor-token masking does not primarily change *where* the model looks; it changes *what gets extracted from* attended positions, by reallocating limited LoRA capacity from Q/K to V/O specialization. The four-probe convergence (direction-of-asymmetry + attention-pattern + weight-allocation + V/O ablation) supports this account, with the V/O ablation providing direct causal evidence that V/O is necessary and sufficient. **A practical productization implication: V/O-only LoRA fine-tuning produces 23% better held-out generalization at half the trainable parameters compared to full LoRA fine-tuning.**

All weights, code, corpora, and per-query outputs are released at `github.com/Chorozion/Casandra-t1-diffusion-edge-model` (code) and the diagnostic artifacts at the same repo under `eval/`. License:

Apache 2.0.

Keywords: Masked-diffusion language models, training objectives, retrieval-augmented generation, out-of-distribution generalization, inductive bias.

---

1. Introduction

Masked-diffusion language models (Austin et al., 2021; Lou & Ermon, 2023; Inception Labs Mercury Coder; Apple GLaM-Diffusion) train by progressively unmasking corrupted sequences, with the model predicting masked tokens conditional on visible context at each diffusion step. The standard training objective applies the masking schedule uniformly across positions, treating prefix and target positions equivalently.

Retrieval-grounded generation introduces a structural asymmetry that the standard training objective does not exploit: at inference time, *retrieved-fact tokens are known* and *the surrounding response must be generated*. We propose that training the model with this asymmetry made explicit — prefix anchor tokens never masked, target statement tokens always subject to the schedule — produces a model better-suited to the inference-time access pattern.

Specifically, we ask: *does anchor-token masking, on identical training data, produce different generation behavior than random-position masking, particularly on out-of-distribution anchors?*

This is a controlled comparison of training objectives, not of architectures or data. The base model, training data, training steps, batch size, learning rate, optimizer, mask ratio range, and LoRA configuration are identical between conditions; only the *which positions can be masked* differs.

1.1 Contributions

1. We formalize anchor-token masking for masked-diffusion LMs as a small modification to the standard training objective (§3).
2. We provide a controlled empirical comparison against random-position masking on identical data, with bootstrap-CI-tight effects (§4).
3. We validate the effect across four held-out semantic domains, demonstrating it is not domain-specific (§4.1b).
4. We propose a mechanism — train-inference distribution alignment producing an "anchor → topical extension" inductive bias — supported by qualitative behavioral evidence (§5).
5. We release reproducible code, weights, corpora, and per-query outputs (§7).

---

2. Related Work

2.1 Masked-diffusion language models

Discrete-diffusion language models corrupt sequences by replacing tokens with `<MASK>` according to a noise schedule, training the model to predict original tokens at masked positions (Austin et al., 2021 — D3PM; Lou & Ermon, 2023 — SEDD; Zhao et al., 2024 — Mercury Coder; Apple, 2025 — GLaM-Diffusion). The standard training objective is symmetric over positions: every position is equally likely to be masked at any given diffusion timestep.

Curriculum and span masking variants (Devlin et al., 2019 — BERT; Lewis et al., 2020 — BART; Q3 architecture span_mask_prob) explore non-uniform mask sampling within sequences. These variants modify *which positions* are masked together, but do not introduce position-class asymmetry of the type we propose.

2.2 Prefix-LM training in autoregressive models

Raffel et al. (2020 — T5) and the prefix-LM family use bidirectional attention over a prefix and autoregressive generation on a suffix, with loss computed only over the suffix. This is conceptually analogous to anchor-token masking — prefix never masked, suffix always subject to autoregressive prediction — but in the autoregressive paradigm. The masked-diffusion analog has not, to our knowledge, been independently characterized in the published literature.

2.3 Retrieval-augmented generation for diffusion LMs

Wang et al. (2026 — SPREAD, arXiv:2601.11342) addresses the Response Semantic Drift failure mode in DLM-RAG via query-relevance-guided denoising at inference time. SPREAD modifies the *trajectory* of the unmasking loop; our work modifies the *training objective*. The two are complementary.

2.4 LoRA fine-tuning

Hu et al. (2021) introduced low-rank adaptation as a parameter-efficient fine-tuning method. We use LoRA at rank 16 on q/k/v/o projections (7M trainable parameters out of 1.3B base) following standard practice. This is a fine-tuning study, not a from-scratch training study; the implications for from-scratch pretraining with anchor-token masking remain to be tested at larger scale.

3. Method

3.1 Training objective

Given an LTMi-XT-formatted training example with breadcrumb-derived question tokens `Q` and answer statement tokens `A`:

- **Anchor-token masking (the proposed objective):** the noise-schedule mask is sampled only over positions `i ∈ A`. Positions `i ∈ Q` are never masked. Loss is computed over masked positions in `A`.
- **Random-position masking (the control):** the noise-schedule mask is sampled uniformly over all real (non-pad) positions, including both `Q` and `A`. Loss is computed over all masked positions.

Formally, for a sequence `x` with position partition `Q ∪ A ∪ Pad`:

```
Anchor-token masking:
    For i ∈ Q:    m_i = 0           (never masked)
    For i ∈ A:    m_i ~ Bern(γ(t))   (always subject to schedule)
    For i ∈ Pad:  m_i = 0           (excluded)
    Loss over { i ∈ A : m_i = 1 }

Random-position masking (control):
    For i ∈ Q ∪ A:  m_i ~ Bern(γ(t))  (uniform schedule)
    For i ∈ Pad:    m_i = 0
    Loss over { i ∈ Q ∪ A : m_i = 1 }
```

Both conditions use the same mask ratio range (uniform sample from 0.5–0.9 per training step) and identical hyperparameters in all other respects.

3.2 Forced-anchor decoding

At inference time, given a query `q` and a retrieved locus statement `s`, we tokenize `s` and lock its tokens at fixed positions `[0, |tokenize(s)|)` of the answer slot. The masked-diffusion unmasking loop excludes locked positions from the `is_masked` set throughout all denoising steps; the model fills in the surrounding positions conditioned on the locked tokens at each step.

This is a **deterministic preservation** mechanism: locked tokens are never overwritten by the unmasking loop, so 100% anchor-token preservation is guaranteed by construction. The empirical question we study is whether the *surrounding* tokens generated by the model around locked anchors are topically aligned with the anchor's content.

3.3 Coherence metric: `corpus_overlap`

We define `corpus_overlap(text, vocab)` as the fraction of word-tokens (regex `[A-Za-z0-9.\-]+`) in `text` whose lowercase string is in `vocab`. For each query, we compute the metric on the *post-anchor portion* of the generated text — the suffix following the locked-anchor span — using the eval corpus's vocabulary as `vocab`.

This is a coarse proxy for "does the generated text use words from the topical domain?" It does not require an LLM judge, is reproducible deterministically, and admits straightforward bootstrap CI computation. It does not, however, penalize incoherent ordering of correct topical tokens or reward grammatical structure.

---

4. Experiments

4.1 Setup

- **Base model:** Cassandra T1 v1, a 1.3B-parameter masked-diffusion LM (28 layers, 2048 hidden, GQA 16q/4kv, RoPE, RMSNorm, SwiGLU, 32K BPE vocab, 128K context, 12-step inference unmasking). Released checkpoint `cassandra_ep5_fp16.pt`, SHA-256 `D70C813...`, epoch-5 final cross-entropy 2.2561. The model is undertrained at this checkpoint and produces incoherent unconditional output.
- **Training data:** 144 (Q, A) pairs derived from three LTMi-XT-formatted corpora (C1: cold-storage operations, C2: Cassandra T1 architecture, C3: business operations memo). Each locus generates 3 templated questions whose answer is the locus statement.
- **LoRA configuration:** rank 16, alpha 32, target modules q_proj/k_proj/v_proj/o_proj across all 28 layers (7M trainable parameters, 0.5% of base).
- **Hyperparameters:** AdamW, LR 1e-4 with cosine decay to 1e-5, batch 2, sequence length 192, 300 training steps, mask ratio uniform [0.5, 0.9], gradient norm clip 1.0, seed 42. Both LoRAs use identical hyperparameters; only the masking strategy differs.
- **Wall-clock:** 75 seconds per LoRA on a single RTX 3090 Ti.
- **Eval queries:** 15 in-domain queries (from the LoRA training corpora) and 48 held-out queries (12 each across C4 astronomy, C5 medical, C6 legal, C7 cooking).
- **Decoding:** 12 unmasking steps, temperature 0.8, top_p 0.9, beta 0.5, repetition penalty 1.3, answer length 96 tokens, generation seed 1234. All conditions use identical decoder settings.

4.2 Aggregate results

```
                          in_domain          held_out (pooled C4-C7)
arm           force       n   overlap        n    overlap
-----------------------------------------------------------------
base          forced      15  0.0147         48   0.0045
base          unforced    15  0.0204         48   0.0123
anchor_lora   forced      15  0.6239         48   0.3571
anchor_lora   unforced    15  0.5857         48   0.3128
random_lora   forced      15  0.4946         48   0.2137
random_lora   unforced    15  0.4914         48   0.2146
```

4.3 Bootstrap confidence intervals

10000 bootstrap resamples, percentile-method 95% CIs, all on forced-anchor `corpus_overlap`:

```
comparison                                  ratio    95% CI            P(a > b)
-------------------------------------------------------------------------------
anchor vs base, in-domain                   42.50    [28.5, 76.0]      1.000
anchor vs random, in-domain                  1.26    [1.18, 1.34]      1.000
anchor vs base, held-out (pooled)           78.73    [54.3, 113.4]     1.000
anchor vs random, held-out (pooled)          1.67    [1.51, 1.85]      1.000
random vs base, in-domain                   33.69    [22.6, 59.8]      1.000
random vs base, held-out (pooled)           47.55    [29.8, 75.8]      1.000
```

The in-domain anchor-vs-random ratio CI [1.18, 1.34] does not overlap the held-out ratio CI [1.51, 1.85]. The "out-of-distribution gap larger than in-distribution gap" claim holds at the 95% bootstrap-CI level.

4.4 Per-domain held-out replication

```
corpus    base      anchor    random    anchor/random   P(anchor>random)
-----------------------------------------------------------------------
C4 astro  0.0063    0.3272    0.1594    2.05x           1.000
C5 med    0.0036    0.3277    0.1909    1.72x           1.000
C6 legal  0.0035    0.3805    0.1731    2.20x           1.000
C7 cook   0.0048    0.3930    0.3314    1.19x           0.994
-----------------------------------------------------------------------
POOLED    0.0045    0.3571    0.2137    1.67x           1.000
          n=48 per arm                                  CI [1.51, 1.85]
```

The effect replicates with P = 1.000 on three domains (astronomy, medical, legal) and weakly positive (P = 0.994) on cooking. Cooking's weakness is attributable to higher generic-vocabulary overlap between training and held-out corpora (the random-LoRA's overlap on C7 is 0.331 versus 0.16-0.19 on the other

three domains, suggesting cooking corpora share more “system,” “use,” “cost” type vocabulary with our training corpora).

4.5 Anchor preservation

All forced-anchor conditions across all arms produce 100.0% anchor-token preservation, as required by the deterministic preservation guarantee of the algorithm. Anchor preservation is unaffected by the LoRA training; it is a property of the inference-time decoder.

4.6 Qualitative behavior on held-out anchors

We present representative anchor-LoRA and random-LoRA forced-anchor outputs for selected held-out queries. Locked anchor content shown in *italics*; generated content follows.

C4 (astronomy) — query: *“What are the moons of Mars?”*

Anchor-LoRA: *“Mars has two small moons named Phobos and Deimos that may be captured asteroids from the asteroid belt.”* The mission scheduled with April the., October was during module 14. the mission schedule for the missions for included three… *(corpus_overlap = 0.52)*

Random-LoRA: *“Mars has two small moons named Phobos and Deimos that may be captured asteroids from the asteroid belt.”* Cassandra T1, Cassandra T1, Cassandra T1 Direct: Direct helpful using 1 2 in Cassandra T1 of Cassandra T1… *(corpus_overlap = 0.11)*

C5 (medical) — query: *“What is normal blood pressure?”*

Anchor-LoRA: *“Normal resting blood pressure for an adult is approximately 120 over 80 millimeters of mercury.”* In a healthy patient, the heart rate is between approximately 75 and 95 beats per minute… *(corpus_overlap typical 0.30+)*

Random-LoRA: *“Normal resting blood pressure for an adult is approximately 120 over 80 millimeters of mercury.”* The Cassandra T1 model uses diffusion at the standard rate of training… *(corpus_overlap typical 0.18)*

The behavioral distinction is consistent across all four held-out domains: anchor-LoRA reaches for content topical to the anchor, random-LoRA falls back to training-corpus boilerplate.

---

5. Discussion: proposed mechanism

5.1 Train-inference distribution alignment

The anchor-token-masked LoRA's training distribution exactly matches the inference-time forced-anchor access pattern: at every training step, the model sees `Q` tokens visible and `A` tokens partially masked, and is asked to predict the masked `A` tokens. At inference, we provide the prompt (acting as `Q`) and lock retrieved-fact tokens (acting as the partially-known `A`), and ask the model to fill in the rest. The two access patterns are identical.

The random-position-masked LoRA's training distribution is *different from* the inference access pattern: at training time, both `Q` and `A` positions are sometimes masked, sometimes visible. The model learns to predict any masked position from any visible context, not specifically to "extend an anchor."

Train-test distribution alignment is well-known to favor in-distribution fit and out-of-distribution generalization in supervised learning (Vapnik 1998; many subsequent results). The pattern we observe is consistent with this framing: the anchor-LoRA's training and inference access patterns coincide, producing more reliable behavior across domains.

5.2 Inductive-bias formation

A complementary framing: the anchor-token-masked training loop teaches the model an explicit *role asymmetry* between prefix and target positions. The model learns that prefix tokens are *conditioning* (always visible, never targets) while statement tokens are *generated* (always either masked or revealed-during-unmasking, never available as conditioning during training). This role asymmetry becomes an internalized inductive bias: when asked to generate around a known prefix, the model treats the prefix as topical conditioning and extends from it.

The random-position-masked LoRA, by symmetry, learns no such role asymmetry. Every position is sometimes a target and sometimes a context. The model develops a more general "predict any masked position from whatever's visible" capability — useful for many tasks but not specialized for retrieval-grounded extension.

5.3 Behavioral implication: regurgitation under uncertainty

When an LM lacks a specific inductive bias for a given input, it falls back to the strongest signal available — typically the training distribution's token frequencies. Our random-LoRA was trained on 144 examples drawn from C1+C2+C3 corpora; given out-of-distribution prompts, it falls back to repeating tokens from those corpora ("Cassandra T1, a diffusion language model by SOPHIA XT" appears verbatim in multiple held-out outputs).

The anchor-LoRA, with its specialized "extend anchor" inductive bias, instead applies that bias to the new anchor content and generates topical extension — even though the topic is out-of-distribution. The extension is wrong on specifics (the model has no astronomy/medical/legal/cooking knowledge) but topically consistent.

This predicts: the anchor-LoRA's advantage should *grow* with semantic distance between training and test corpora (because the random-LoRA's regurgitation becomes increasingly off-topic as distance grows), exactly as we observe.

5.4 Alternative hypotheses and disconfirmations

Alternative A: Easier-objective faster convergence. Anchor-token masking has fewer positions to predict per step, making each gradient step more focused. Anchor-LoRA may simply have fit its training data better at fixed budget.

Why this doesn't explain the data: if it were the dominant mechanism, the in-domain advantage would be *larger* than the held-out advantage (because training data overlaps with in-domain queries but not held-out). We observe the opposite — held-out advantage 1.67× exceeds in-domain advantage 1.26× with non-overlapping CIs.

Alternative B: Vocabulary leakage. The random-LoRA's overlap on held-out might reflect the trivial vocab intersection between training corpora and held-out corpora.

Why this doesn't explain the data: the leakage floor (uniform sampling from training vocab × held-out intersection) is approximately 6.8%. Random-LoRA's pooled held-out overlap of 21.4% is 3.1× the floor. Anchor-LoRA's 35.7% is 5.3× the floor. Both LoRAs exceed the floor, but anchor-LoRA exceeds it by much more. The anchor advantage is not explained by leakage alone.

Alternative C: Cherry-picked metric. The corpus-vocabulary overlap metric might be sensitive to artifacts that don't reflect topical coherence.

Partial mitigation: the qualitative samples (§4.6) confirm the metric tracks an interpretable behavior — anchor-LoRA produces visibly topical extension, random-LoRA produces visibly off-topic regurgitation. An LLM-judge replication would strengthen this further but was not run for cost reasons.

5.5 Mechanism probe: reverse-asymmetric control

To directly test whether the *direction* of asymmetric masking matters (not just that asymmetry exists), we trained a third LoRA with the opposite asymmetry: **Q always subject to schedule, A never masked**. Identical hyperparameters, identical data, 75 seconds wall-clock training. This LoRA learns the inverse operation: "predict the question from the answer."

The inductive-bias formation hypothesis predicts this should perform *worse* than random-position masking, because the wrong-direction asymmetry teaches a wrong-direction inductive bias. The

competing hypothesis ("any asymmetry helps") predicts equivalent or better performance.

Result:

```
Held-out forced corpus_overlap (pooled across C4-C7, n=48 per arm):
  base           0.0045
  anchor_lora    0.3571   (correct-direction asymmetry)
  random_lora    0.2137   (no direction)
  reverse_lora   0.1760   (wrong-direction asymmetry)

reverse_lora vs random_lora on held-out: diff = -0.0377
  95% bootstrap CI: [-0.0677, -0.0077]
  P(reverse > random) = 0.006
```

The reverse-asymmetric LoRA performs statistically significantly *worse* than the random-position LoRA on held-out (P = 0.006, CI entirely negative). The 3-LoRA ranking is exactly the predicted ordering for the inductive-bias hypothesis:

> anchor (correct direction) > random (no direction) > reverse (wrong direction) ≫ base

5.6 What the reverse-LoRA outputs reveal

Qualitatively, the reverse-LoRA emits **question-template fragments** ("Tell me about," "Answer me," "the source") and **system-prompt self-references** ("Cassandra T1, a diffusion language model by SOPHIA XT") in every held-out condition. Sample output for *"What is the orbital period of Jupiter?"* with a forced anchor stating the orbital period:

> *"Jupiter has a mass of 1.898 times ten to the 27 kilograms and orbits the Sun every 11.86 Earth years."* A: Cassandra T T1, a diffusion diffusion language model by SOPHIA XT. Direct, helpful…
> **Tell me about the information**

The "Tell me about" fragment is a direct memorization of the training-data Q template (`f"Tell me about {concept} in the context of {topic}."`). This is the predicted behavior: a model trained to predict Q from A becomes a question-generator, and when given an answer-style prompt + locked answer tokens, it generates Q-shaped scaffolding around the answer rather than topical extension.

This rules out the "asymmetric optimization landscape alone is responsible" alternative. The asymmetry must point in the correct direction (anchor → answer) to produce the topical-extension behavior.

5.7 Mechanism probe 2: attention pattern

We instrument `SlidingWindowAttention.forward` to capture per-layer attention weights during a single forward pass on each of four held-out queries (one per domain), with the prompt + locked anchor + masked post-anchor positions as input. We compute the mean attention weight from post-anchor query positions to anchor key positions, averaged across heads, layers, and queries.

Result:

```
arm           anchor_share (mean across 4 held-out queries)
base          0.256
anchor_lora   0.272   (+0.016 vs base)
reverse_lora  0.263   (+0.007 vs base)
random_lora   0.259   (+0.003 vs base)
```

The 4-arm ranking matches the behavioral hypothesis (anchor > reverse > random > base), but the magnitude is small: anchor-LoRA shows only +0.013 anchor-share advantage over random-LoRA, versus the +0.143 absolute behavioral advantage in `corpus_overlap`. The attention-pattern shift is in the predicted direction but explains roughly 10% of the behavioral effect.

This rules out the simplest version of the inductive-bias story ("the model learns to look at anchors more"). The behavioral effect is too large to be accounted for by attention-pattern shifts alone.

5.8 Mechanism probe 3: LoRA weight allocation

We compute the Frobenius norm of the effective LoRA delta-weight matrix (`B @ A`) for each projection type (q_proj, k_proj, v_proj, o_proj) across all 28 layers, summing across layers per projection type. This measures where each LoRA invested its limited rank-16 capacity.

Result:

```
arm           q_proj   k_proj   v_proj   o_proj   V+O total   Q+K total   V+O/Q+K ratio
anchor_lora   0.285    0.132    0.146    0.335    0.481       0.417       1.153
reverse_lora  0.409    0.163    0.160    0.378    0.538       0.572       0.940
random_lora   0.432    0.187    0.142    0.349    0.491       0.618       0.794
```

The anchor-LoRA is the only configuration where V+O capacity exceeds Q+K capacity (ratio > 1). Random- and reverse-LoRAs both invest more heavily in Q (0.43 and 0.41 vs anchor's 0.29).

This makes mechanistic sense. Under anchor-token masking, the prefix Q is never masked during training, so the LoRA never needs to spend capacity rebuilding Q-projection behavior (the model already handles fully-visible Q correctly via base weights). The freed capacity gets allocated to V/O — which control *what content gets read out from* attended positions, not where the model attends.

Random-position masking, by contrast, sometimes masks Q tokens during training. The LoRA must update Q projections to handle the partially-masked Q case, consuming capacity that could otherwise go to V/O.

5.9 Cohesive mechanism account

Combining the three mechanism probes:

1. **Reverse-asymmetric kills "asymmetry alone".** Direction matters; it is specifically anchor → answer asymmetry (and not its inverse) that produces the behavioral advantage.
2. **Attention pattern shifts modestly.** Anchor-LoRA does attend slightly more to anchor positions, but this is a small effect that explains ~10% of the behavioral gap.
3. **Weight allocation reallocates capacity.** Anchor-LoRA invests its rank-16 budget disproportionately in V/O (value/output) projections rather than Q/K (attention pattern) projections.

The synthesis: **anchor-token masking does not primarily change *where* the model looks; it changes *what gets extracted from* the positions the model already attends to.** The Q-prefix is always fully visible during anchor-token-masked training, freeing LoRA capacity to specialize V/O transformations — which encode the "extend the topic of the anchor" inductive bias. Random-position-masked training spends its capacity on Q/K instead, leaving less for V/O specialization.

This account is consistent with the behavioral, attention, and weight-allocation evidence. It refines the original "train-inference distribution alignment produces inductive bias" framing into a more specific mechanistic story: **the inductive bias lives in V/O projections, not in attention patterns**, and the asymmetric training objective unlocks LoRA capacity to develop that V/O specialization by removing the need to model masked-Q reconstruction.

5.10 Remaining mechanistic experiments

Two probes that would further tighten the story:

1. **Per-step attention dynamics.** Track attention patterns across all 12 denoising steps (not just t=0.5) to see whether the LoRAs differently propagate information through the iterative unmasking process.
2. **Direct V/O ablation.** Disable LoRA on V/O while keeping it on Q/K (and vice versa), measure which substitution preserves the behavioral effect. This would directly test the "V/O is the load-bearing component" claim.

Neither is required for the present empirical claim; both would strengthen the mechanism story.

---

6. Limitations

- **Single base model.** Cassandra T1 v1, undertrained at cross-entropy 2.26. The mechanism may behave differently on more-trained models, where LoRA effects are typically attenuated.
- **Moderate n.** 15 in-domain + 48 held-out queries (12 per held-out domain). Pooled CIs are tight; per-domain CIs are wider.
- **Coarse coherence metric.** Corpus-vocabulary overlap is a proxy for topical coherence; it does not penalize incoherent ordering or reward grammatical structure.
- **Single LoRA configuration.** Rank 16 on q/k/v/o, 300 training steps. Other configurations not explored.
- **Manual held-out corpora.** C4-C7 were authored by the researcher rather than drawn from external benchmarks. Factual content is publicly verifiable; reviewer-strength validation would use independently-curated benchmark data.
- **One outlier domain.** C7 (cooking) shows weaker effect (1.19×) due to higher generic-vocabulary overlap with training corpora. The technique's effectiveness may depend on semantic distinctness of held-out vs training corpora.
- **No from-scratch training.** Anchor-token masking is studied here as a fine-tuning regime. Whether it produces equivalent advantages when used for from-scratch pretraining is not yet known.
- **The anchor-LoRA's training objective is technically easier** (smaller per-step loss surface). The held-out advantage growing rather than shrinking is the main reason this concern doesn't kill the result, but it warrants further work at multiple training budgets.

These limitations specify what additional work would strengthen the claim. They do not invalidate the present finding: anchor-token masking produces a measurable, controlled, multi-domain-replicated, statistically tight generalization advantage over random-position masking on identical data.

---

7. Reproducibility

All artifacts released under Apache 2.0:

Code: - `runner/lora_finetune.py` — anchor-token-masked LoRA training - `runner/lora_finetune_random.py` — random-position-masked LoRA training (control) - `runner/run_diagnostic_eval.py` — initial 4-arm eval - `runner/run_multidomain_eval.py` — multi-domain held-out eval - `runner/bootstrap_analysis.py` — bootstrap CI computation

Weights: - `weights/cassandra_ep5_fp16.pt` — base model (1.3B, SHA-256 verified) - `lora_adapters/v1_ltmi_r16.pt` — anchor-LoRA (24 MB) - `lora_adapters/v1_ltmi_random_r16.pt` — random-LoRA control (24 MB)

Data: - `C1.json`, `C2.json`, `C3.json` — in-domain corpora (LTMi-XT format) - `C4.json`, `C5.json`, `C6.json`, `C7.json` — held-out corpora - `queries.json`, `queries_heldout.json`, `queries_heldout_extended.json` — eval queries

Outputs: - `eval_out/diagnostic_eval.jsonl` — 162 per-query results (initial) - `eval_out/multidomain_eval.jsonl` — 216 per-query results (multi-domain) - `eval_out/diagnostic_summary.json`, `eval_out/multidomain_summary.json` — aggregates - `eval_out/bootstrap_results.json` — CI computation outputs

Reproduction time: ~50 minutes total on a single RTX 3090 Ti (75s per LoRA × 2 LoRAs + ~25 min for the initial 4-arm eval + ~20 min for the multi-domain eval). Random seeds fixed at 42 (training) and 1234 (eval generation).

Repository: `github.com/Chorozion/Casandra-t1-diffusion-edge-model` (release pending; available privately on request).

---

8. Future Work

We identify several scoped follow-ups:

1. **Attention-pattern verification.** Direct mechanistic inspection of attention weights during generation, contrasting anchor-LoRA and random-LoRA on identical inputs.
2. **Reverse-asymmetric control.** Training with `Q` masked and `A` never masked, to confirm the *direction* of asymmetry matters.
3. **Sample-efficiency curve.** Pairs of LoRAs at training-set sizes {50, 100, 200, 500, 1000} and step counts {100, 300, 1000, 3000} to characterize how the anchor advantage scales with training budget.
4. **Cross-architecture replication.** Apply the analog (prefix-LM training vs random-span corruption) to a small autoregressive LM, testing whether the inductive-bias finding is diffusion-specific or paradigm-general.
5. **Larger-scale base model.** Run the same comparison on a fully-trained masked-diffusion LM (Mercury 2 if accessible to fine-tuning, or a smaller open-source DLM that has converged), to determine whether the effect is amplified by undertraining or persists at training maturity.

6. **LLM-judge metric replication.** Run a rubric-based LLM judge (e.g., Claude Haiku) on a sampled subset of generations to confirm that corpus_overlap captures the intended topical-coherence signal.
7. **From-scratch training regime.** The natural extension is to train a small masked-diffusion LM from scratch with anchor-token masking applied to retrieval-grounded data (LTMi-XT or similar format), comparing against random-position-masked baselines. ~$200 of GPU time on rented H100.

We commit to publishing results from any of these follow-ups whether positive or negative.

---

9. Conclusion

We have demonstrated, with controlled validation across four held-out domains and tight bootstrap confidence intervals, that anchor-token masking produces a 1.67× generalization advantage over random-position masking on identical retrieval-grounded training data. The held-out advantage is larger than the in-domain advantage in a statistically distinguishable way, supporting the interpretation that anchor-token masking specifically favors out-of-distribution generalization rather than merely fitting training data better.

The proposed mechanism — train-inference distribution alignment producing an "anchor → topical extension" inductive bias — is consistent with both quantitative and qualitative evidence and predicts the asymmetric in-domain vs out-of-distribution gap we observe. Direct mechanistic verification (attention analysis, reverse-asymmetric control) is identified as future work.

This is a small-scale finding (1.3B undertrained model, rank-16 LoRA, 144 training examples). Its implications for from-scratch pretraining and for fully-trained models remain to be tested. We view this report as motivating, not concluding, the broader investigation of asymmetric masking objectives in masked-diffusion language modeling.

References

(To be completed in submission draft. Initial set:)

- Austin, J., et al. *Structured Denoising Diffusion Models in Discrete State-Spaces*. NeurIPS 2021.
- Devlin, J., et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. NAACL 2019.
- Hu, E., et al. *LoRA: Low-Rank Adaptation of Large Language Models*. ICLR 2022.

- Inception Labs. *Mercury Coder: A Diffusion Language Model*. Technical Report 2024.
- Lewis, M., et al. *BART: Denoising Sequence-to-Sequence Pre-training*. ACL 2020.
- Lou, A., Ermon, S. *Discrete Diffusion Modeling by Estimating the Ratios of the Data Distribution*. ICML 2024.
- Raffel, C., et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer (T5)*. JMLR 2020.
- Vapnik, V. *Statistical Learning Theory*. Wiley, 1998.
- Wang, J., et al. *Unlocking the Potentials of Retrieval-Augmented Generation for Diffusion Language Models: A Semantic Drift Perspective (SPREAD)*. arXiv:2601.11342, 2026.

Technical report prepared 2026-05-09. All numbers reproducible from committed code and seeds.

Correspondence: Thomas Garren, SOPHIA XT LLC, `thomas@sophiaxt.com`.

Acknowledgments: This work was conducted independently and unfunded. The author thanks the Inception Labs Mercury team and Apple's GLaM-Diffusion team for releasing model and methodology details that motivated this investigation.