Reviser: Revision-Capable Text Generation via Autoregressive Cursor Actions

Anonymous ACL submission

Abstract

Revision-capable generation is appealing because it can insert or revise earlier content, but many non-autoregressive and edit-based approaches obtain this flexibility through repeated sequence-level computation. We propose Reviser, a decoder-only Transformer that generates a response as a sequence of cursor-relative actions on a mutable canvas. At each step, Reviser predicts exactly one action token: INSERT(token), MOVE(Δ), or STOP, and is autoregressive over edit-history actions rather than final text order.

This design enables genuinely non-monotonic generation while preserving a simple next-action interface. On a continuation benchmark, Reviser is strongly preferred to SEDD and MDLM in our arena evaluations, and trajectory statistics confirm that the model performs frequent backward moves and mid-canvas insertions rather than merely emulating end-append decoding. Against size-matched autoregressive baselines, Reviser is competitive at both the 100M and 300M scales. Under our shared FLOPs convention, Reviser also requires substantially less inference compute than representative multi-pass refinement and diffusion-style baselines.

1 Introduction

Autoregressive (AR) Transformers dominate text generation but are structurally biased toward producing content in final left-to-right order: once a clause is emitted, correcting it typically requires appending more text rather than directly revising earlier content (Vaswani et al., 2017; Brown et al., 2020). Revision-capable and non-autoregressive (NAR) methods promise post-hoc correction and parallelism, yet often incur large overhead—multiple full-sequence refinement passes, scoring every insertion slot, or many-step sampling (Ghazvininejad et al., 2019; Gu et al., 2019; Li et al., 2022; Lou et al., 2024; Sahoo et al., 2024). We ask: can we obtain revision-style generation while keeping total compute close to a standard AR Transformer?

We propose **Reviser**, which generates by executing a stream of cursor edits on a mutable canvas. The model is a standard decoder-only Transformer, but it is *autoregressive over action tokens (the edit history), not over the final text token order*. Because the cursor can move and insert into earlier positions, Reviser is not constrained to left-to-right generation in the final text order, enabling revision-like behavior while keeping each model call "normal-sized." In our primary instantiation Reviser uses only INSERT and MOVE actions (plus STOP), where "revision" means non-left-to-right insertion into earlier positions; the same deterministic executor naturally supports richer operators (DELETE/REPLACE/span edits; Appendix A.1), which we leave to future work.

Rather than outputting a distribution over all positions (or all insertion slots) at each step, Reviser outputs exactly one cursor-relative action token per step. Crucially, the Transformer trunk attends only to the action-history sequence H_t; it never attends to canvas tokens directly. The canvas is implicit in the history of executed actions and is accessed only through validity masking. Each model call is therefore comparable to a standard AR step (Transformer trunk + one head); the only remaining overhead is the number of action steps, which is close to the output length in an insert-dominant regime.

A concrete example illustrates the behavior; we write the cursor as a vertical bar "|" between tokens. In one 300M Reviser generated trajectory (Appendix E):

Before edit. `Information is deemed correct and is subject to change. Real estate listings obtained from third party sources are for | consumers' personal purchasing decisions and should…`

Actions (steps 204–209). MOVE(-16) + INSERT(`at the time of publishing`)

After. `Information is deemed correct at the time of publishing | and is subject to change. Real estate listings obtained from third party sources are for…`

This edit makes the statement more precise; Appendix E provides full qualitative examples and trajectories.

Contributions. (i) We introduce Reviser, a cursor-action generator that is autoregressive over edit-history actions rather than final text order. (ii) We formalize the canvas state, cursor edit operators, validity masking, and the deterministic executor for the insert+move+STOP setting. (iii) We describe an obfuscation–restoration scheme for supervising next-action predictors on edit trajectories. (iv) We show empirically that Reviser produces genuinely non-monotonic trajectories, is strongly preferred to SEDD/MDLM and competitive with size-matched AR baselines at 100M and 300M, and—under a shared FLOPs convention—is substantially cheaper than representative multi-pass refinement and diffusion-style baselines.

2 Problem Setup and Design Goals

Reviser supports prompt-conditioned generation in general; here we instantiate it as a prefix-seeding text-continuation model: given an input prefix $x = (x_1, \ldots, x_m)$, generate the continuation. To make the prefix visible to the history-only model, we write it into both the canvas and the action history via a deterministic insert-only prefix-seeding sequence $\mathfrak{A}^{\text{pref}}(x) = (\text{INSERT}(x_1), \ldots, \text{INSERT}(x_m))$. Executing $\mathfrak{A}^{\text{pref}}(x)$ from a blank canvas places each prefix token in order, sets the cursor at position m, and makes the prefix visible through H_t; continuation decoding then begins from this seeded state, with a validity mask that prevents edits to the prefix region.

Our design goal is revision-capable generation—the ability to insert or modify earlier content during decoding—while keeping the compute profile close to a standard AR Transformer. We aim to avoid common NAR overheads: (i) repeated full-sequence refinement passes, (ii) per-step scoring over all positions/slots, (iii) large per-position action heads, and (iv) samplers requiring many denoiser steps. Reviser instead makes a single cursor-relative decision per step using a standard trunk and one action head. We evaluate (a) quality (evalPPL and arena win rates), (b) compute (FLOPs), and (c) trajectory behavior (steps/token, move fraction, cursor travel, insertion locations). Table 1 summarizes notation.

Symbol	Meaning
x, m	input prefix and its length
$\mathfrak{A}$	restoration trajectory (action sequence)
C_t, ℓ_t	canvas tokens and length at step t
$u_t \in \{0, \ldots, \ell_t\}$	cursor boundary index
H_t	edit-history actions at step t
V_c, V_a	token / action vocabulary
n	final output length (tokens)
p_{move}	fraction of actions that are moves
n_{eff}	effective length $n/(1 - p_{\text{move}})$
T_{rest}	restoration length (incl. STOP)
$T_{\text{dec}}, T_{\text{mp}}$	diffusion / Mask-Predict iters
$I^{\mathcal{M}}_{\text{infer}}$	inference FLOPs multiplier vs. AR

Table 1: Core notation used throughout.

3 Compute Tradeoffs in NAR and Edit-Based Transformers

Many NAR and edit-based Transformers obtain flexibility or parallelism by repeating expensive sequence-level computation multiple times per response (Lee et al., 2018; Ghazvininejad et al., 2019; Gu et al., 2019; Li et al., 2022; Lou et al., 2024; Sahoo et al., 2024). Under a shared FLOPs convention this often yields far higher total inference compute than a standard AR baseline. We report the multiplier

$$I^{\mathcal{M}}_{\text{infer}} \triangleq \frac{F^{\mathcal{M}}_{\text{infer}}}{F^{\text{AR}}_{\text{infer}}},$$

where $F^{\mathcal{M}}_{\text{infer}}$ is the total FLOPs to generate a length-n response under method $\mathcal{M}$ (full derivations in Appendix B). An AR baseline produces a length-n response with one (KV-cached) full-sequence evaluation plus a vocab head per token. Most NAR families instead run R full-sequence evaluations (refinement it-

erations, insertion rounds, or denoising steps), multiplying both the dominant trunk cost and the large vocab-head cost ($|V| \approx 50$k) by roughly R; e.g., CMLM commonly uses $R \approx 10$ and diffusion-style decoders use $R = T_{\text{dec}} \in [200, 2000]$.

The most compute-efficient NAR variants predict many (or all) tokens at once, but this creates a coordination problem: each position must agree with the unknown choices at other positions. One-shot NAR behaves like a product-of-marginals approximation $p(y \mid x) \approx \prod_{i=1}^{n} p(y_i \mid x)$ (Gu et al., 2018), which cannot reliably enforce inter-token constraints, manifesting as agreement errors, repetition, and contradictions, and motivating additional passes that raise compute.

Reviser avoids both issues: it predicts one cursor-relative action at a time conditioned on the full prior edit history (keeping global coherence through AR dependence while allowing non-monotonic edits), and each step is a standard next-action pass rather than a repeated full-sequence re-evaluation. If a fraction p_{move} of actions are moves, producing n tokens takes an expected $n_{\text{eff}} = n/(1 - p_{\text{move}})$ steps, so

$$I_{\text{infer}}^{\text{Reviser}}(n) \approx \frac{F_{\text{infer}}^{\text{AR}}(n_{\text{eff}})}{F_{\text{infer}}^{\text{AR}}(n)}.$$

For $p_{\text{move}} \in \{0.20, 0.28, 0.33\}$ at $n = 128$, $n_{\text{eff}} \in \{160, 177.8, 191.0\}$ and $I_{\text{infer}}^{\text{Reviser}} \in \{1.25, 1.39, 1.50\}$. Table 2 summarizes the multipliers: with few exceptions, popular NAR Transformers need an order of magnitude or more compute than AR.

Method	Decoding	$I_{\text{infer}}^{\mathcal{M}}$	Par.?
AR (baseline)	n steps	1.00	No
Reviser (ours)	T_{rest} steps	1.25–1.50	No
LevT (edit/refine)	$R \in \{5, 10\}$	6.91–19.40	Yes
InsT (bal. tree)	$\log_2 n$	2.02	Yes
InsT (serial)	n	65.01	No
CMLM (T_{mp}=10)	10	11.86	Yes
Diffusion-LM	200–2000	140.51–1,402.36	No
One-shot NAT	1 enc+1 dec	1.96	Yes

Table 2: Analytic inference-compute multipliers under a shared FLOPs convention ($I_{\text{infer}}^{\mathcal{M}} \triangleq F_{\text{infer}}^{\mathcal{M}}/F_{\text{infer}}^{\text{AR}}$; structural estimates, not measured wall-clock). "Par.?" marks families whose highly parallel variants underperform from global coordination constraints. Reviser's range uses $p_{\text{move}} \in \{0.20, 0.28, 0.33\}$ at n=128. Derivations in Appendix B.

4 Related Work

Reviser is closest to non-autoregressive and partially autoregressive generation that relaxes strict left-to-right decoding. Early one-shot NAT improves parallelism but faces multimodality and coordination issues (Gu et al., 2018); iterative variants such as deterministic refinement and Mask-Predict improve quality by repeating full-sequence passes (Lee et al., 2018; Ghazvininejad et al., 2019). It is also related to insertion- and edit-based generation: the Insertion and Levenshtein Transformers realize flexible generation orders via insertion/deletion (Stern et al., 2019; Gu et al., 2019), and text-edit taggers such as LaserTagger and FELIX frame generation as edit prediction over an existing sequence (Malmi et al., 2019; Mallinson et al., 2020), though many of these were introduced and evaluated mainly in sequence-transduction settings (machine translation and grammatical-error correction) rather than open-ended continuation. PIE-style post-editing makes a similar design choice by treating generation as targeted rewriting rather than pure left-to-right continuation. Reviser is in the same family but uses explicit cursor actions with deterministic executor semantics and keeps a one-action interface per step.

Diffusion-style LMs provide another route to non-monotonic generation, decoupling final token order from AR factorization through denoising trajectories (Li et al., 2022; Lou et al., 2024; Sahoo et al., 2024); more recent semi-AR/block-diffusion variants (LLaDA, Dream, SDLM; analyzed in Appendix B) mix iterative denoising with partial AR structure. Latent-variable NAR models such as FlowSeq relax left-to-right constraints via invertible latents (Ma et al., 2019). Finally, non-left-to-right autoregression includes XLNet's permutation objective over factorization orders (Yang et al., 2019) and insertion-based AR (Stern et al., 2019); Reviser differs by making the edit history itself the autoregressive object—one cursor-relative action per step, executed immediately on a mutable canvas.

Our positioning is therefore narrow: Reviser uses cursor-relative actions over a mutable canvas, predicts exactly one next action token at

a time, and applies edits through a deterministic executor. This preserves revision capability while keeping the decoding interface close to standard next-token prediction and, empirically, much closer to AR compute than multi-pass refinement families.

5 Reviser: State, Actions, and Executor

Reviser maintains a mutable canvas and a cursor indicating the insertion boundary. At each step it reads the history of past actions, applies a validity mask derived from (C_t, u_t), and predicts one next action token (INSERT/MOVE/STOP); a deterministic executor then updates the canvas and cursor. In prefix-seeding, the canvas and history are pre-seeded with $\not{\prec}^{\text{pref}}(x)$. Figure 1 illustrates one step.

State. A generation state at step t is $s_t = (C_t, u_t, H_t)$, where $C_t = (c_{t,1}, \ldots, c_{t,\ell_t})$ is the canvas (length ℓ_t), $u_t \in \{0, \ldots, \ell_t\}$ is the cursor boundary (0 means before the first token), and H_t is the executed action sequence, with $H_1 = \not{\prec}^{\text{pref}}(x)$ and $H_t = (\not{\prec}^{\text{pref}}(x), a_1, \ldots, a_{t-1})$.

Action space. The action vocabulary is $V_a = V_c \cup \mathcal{M} \cup \{\text{STOP}\}$: each token in V_c corresponds to an INSERT action, $\mathcal{M}$ is a small discrete set of moves (e.g., $\Delta \in \{\pm 1, \pm 2, \pm 4, \ldots\}$), and STOP ends generation.

Executor. For a token $x \in V_c$ inserted at boundary u_t,

$$\text{INSERT}(x): \; C_{t+1} = (c_{t,1}, \ldots, c_{t,u_t}, x, c_{t,u_t+1}, \ldots, c_{t,\ell_t}),$$

with $u_{t+1} = u_t + 1$. For a displacement $\Delta \in \mathcal{M}$, MOVE(Δ) leaves $C_{t+1} = C_t$ and sets $u_{t+1} = u_t + \Delta$; STOP terminates and outputs C_t. The post-state constraint $u_{t+1} \in \{0, \ldots, \ell_{t+1}\}$ is enforced by masking.

Validity masking. From (C_t, u_t) we build a valid-action mask that forbids moves with $u_t + \Delta \notin \{0, \ldots, \ell_{t+1}\}$, forbids inserting once ℓ_t reaches a maximum length, and optionally forbids reserved tokens. The mask is applied to the action logits before sampling/argmax.

Algorithm 1 Reviser decoding.

Require: prefix $x = (x_1, \ldots, x_m)$ (possibly empty); max steps $T_{\max}$; max canvas $L_{\max}$
Phase 1 (prefix seeding): $C \leftarrow ()$, $u \leftarrow 0$, $H \leftarrow ()$
for $i = 1, \ldots, m$ **do**
  INSERT(x_i): $C \leftarrow (C, x_i)$, $u \leftarrow u+1$, $H \leftarrow (H, \text{INSERT}(x_i))$
end for
$C_1 \leftarrow C$, $u_1 \leftarrow m$, $H_1 \leftarrow H$
Phase 2 (continuation decoding):
for $t = 1, 2, \ldots, T_{\max}$ **do**
  $r_t \leftarrow f_\theta(H_t)$; $m_t \leftarrow \text{MASK}(C_t, u_t, L_{\max}, |x|)$
  $a_t \sim \text{softmax}(r_t + m_t)$
  if $a_t = \text{STOP}$ **then return** C_t
  else
    $(C_{t+1}, u_{t+1}) \leftarrow \text{EXEC}(C_t, u_t, a_t)$
    $H_{t+1} \leftarrow (H_t, a_t)$
  end if
end for
return $C_{T_{\max}+1}$ ▷ fallback if no STOP

6 Model

Let H_t be the edit-history action tokens at step t. We use a single shared embedding table $E \in \mathbb{R}^{|V_a| \times d}$ for all action tokens (token-valued INSERT actions, MOVE actions, and STOP) and a positional table $P^{(H)}$, so $e_\tau = E[a_\tau] + P^{(H)}[\tau]$. A standard causal Transformer maps $(e_1, \ldots, e_{t-1})$ to hidden states, and the next-action logits are $r_t = W_{\text{out}} h_{t-1} + b_{\text{out}}$ with

$$p(a_t \mid H_t, C_t) = \text{softmax}(r_t + m_t),$$

where m_t is the validity mask from (C_t, u_t) (invalid actions get $-\infty$). The trunk depends only on H_t; the canvas enters only through the executed-action stream and the mask. Optional canvas-conditioning (a pooled summary or cross-attention) is described in Appendices A.2 and A.3 but unused in our runs.

In prefix-seeding, the prefix enters exclusively through the seeded history: the m actions $\not{\prec}^{\text{pref}}(x)$ occupy the first m positions of H_t, so self-attention attends to the full prefix with no separate encoder, and a validity mask prevents editing the prefix region. Algorithm 1 gives the decoding loop: Phase 1 (prefix seeding) is deterministic and makes no model calls; Phase 2 samples continuation actions one at a time until STOP (or a step budget).

7 Learning and Training

Reviser is trained to predict the next action token under teacher forcing on action trajectories. Training shares the two-phase structure

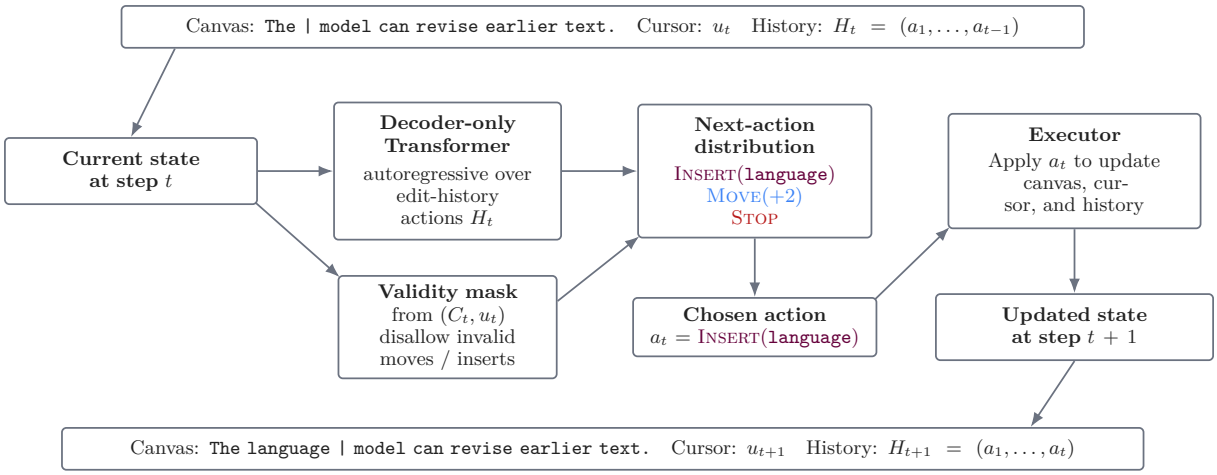

Figure 1: Reviser decoding at a single step. The model is autoregressive over edit-history actions rather than final text order. At each step it predicts one cursor-relative action, applies a validity mask derived from the current canvas state, and uses a deterministic executor to update the canvas, cursor, and history.

of decoding: a deterministic prefix-seeding phase $\mathcal{A}^{\text{pref}}(x)$ followed by a learned restoration phase $\mathcal{A}^{\text{rest}}(y)$ over the continuation target. The negative log-likelihood

$$\mathcal{L}(\theta) = -\sum_{t=1}^{T} \log p_\theta(a_t \mid s_t)$$

is computed only over $\mathcal{A}^{\text{rest}}(y)$ (the prefix is not edited). We render the cursor as a vertical bar "|" between tokens; Figure 2 shows a sample non-monotonic trajectory and Table 3 a worked obfuscation–restoration pair.

Obfuscation–restoration supervision. We construct training trajectories by corrupting a target and inverting the corruption (Algorithm 2; a worked side-by-side example is in Table 11). Initialize the editable canvas to the continuation target and the cursor at a uniformly random boundary: $C \leftarrow y$, $u \sim \text{Uniform}(\{0, \ldots, |y|\})$. At each obfuscation step we sample, with validity masking, DELETE (probability 0.8), which deletes the token immediately left of the cursor, or MOVE(Δ) (probability 0.2) with Δ uniform over valid moves; deleted tokens are recorded in temporal order. Moves use powers-of-two displacements $\mathcal{M} = \{\pm 2^k : 2^k \leq \text{max_move}\}$. Obfuscation terminates when the canvas is empty; trajectories that fail to empty within $K_{\max}$ steps are discarded and resampled.

Given the obfuscation sequence B and the deleted-token stack $\mathcal{D}$, the restoration trajectory $\mathcal{A}$ is built by scanning B last-to-first and inverting each action: each DELETE becomes INSERT(x) with $x = \text{pop}(\mathcal{D})$ (LIFO order), and each MOVE(Δ) becomes MOVE($-\Delta$); we then append STOP. This strict inverse yields one restoration action per obfuscation action ($T_{\text{rest}} = n_{\text{obf}}+1$), and executing $\mathcal{A}$ from a blank canvas deterministically reconstructs the target. Because the construction reverses executor dynamics, every training target is a valid next edit for the current canvas/cursor state rather than a label detached from state, which yields dense, state-consistent edit-history supervision using the same interface as inference.

We use supervised learning only (teacher forcing on restoration trajectories), without DAgger or RL. Both are natural extensions—for compounding error and for sequence-level rewards, respectively—and are described in Appendices A.4 and A.5.

8 Experiments

Setup. We evaluate on a C4 continuation benchmark built from the English validation split of `allenai/c4` (Raffel et al., 2020), using a 35-token prefix and a 180-token total target (examples pre-filtered to total GPT-2 length 144–216). We report 100M and 300M Reviser checkpoints. For the diffusion comparison we use three seeds (`123/124/125`, 1000 prompts each; 3000 total) and decode SEDD/MDLM with 128 steps; Reviser decodes with at most 256 actions. Models we train (on FineWeb; Penedo et al., 2024) use matched 30B-token budgets (AR on text to-

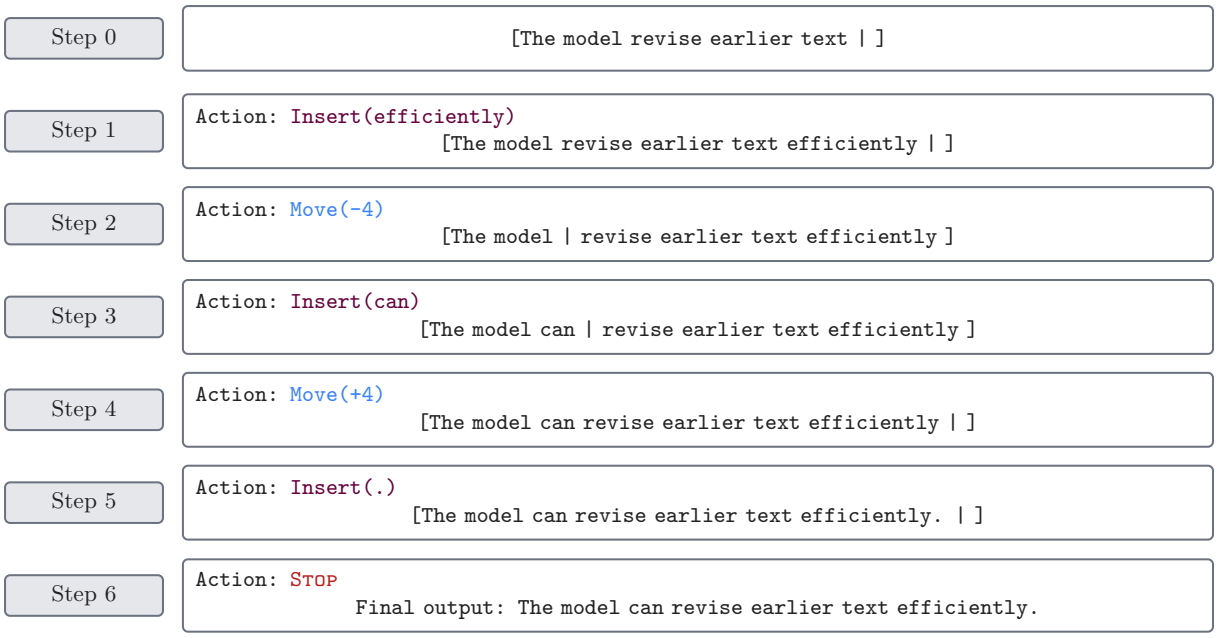

Figure 2: Illustrative Reviser trajectory. The model need not generate in final left-to-right order: it first appends a word to the end (Step 1), moves the cursor backward (Step 2), inserts a word into the middle (Step 3), returns to the end (Step 4), appends a period (Step 5), and stops (Step 6). This backward-move-then-mid-canvas-insertion pattern is common in large-scale trajectory statistics, not an edge case.

Algorithm 2 Obfuscation–restoration trajectory.

Require: target $y = (y_1, \ldots, y_n)$; move set $\mathcal{M}$; max steps $K_{\max}$
$C \leftarrow y$; $u \sim \textsc{Uniform}(\{0, \ldots, n\})$; $B \leftarrow ()$; $\mathcal{D} \leftarrow ()$
for $k = 1, \ldots, K_{\max}$ **do**
  if $|C| = 0$ **then break**
  end if
  sample $b_k \sim \pi_{\text{obf}}$ ($\textsc{Delete}$ w.p. 0.8, else $\textsc{Move}(\Delta)$), masked
  if $b_k = \textsc{Delete}$ **then**
    $x \leftarrow c_u$; delete x; $u \leftarrow u - 1$; push x to $\mathcal{D}$
  else $u \leftarrow u + \Delta$
  end if
  append b_k to B
end for
$\mathfrak{A} \leftarrow ()$
for $k = |B|, |B| - 1, \ldots, 1$ **do**
  if $b_k = \textsc{Delete}$: append $\textsc{Insert}(\text{pop}(\mathcal{D}))$; else append $\textsc{Move}(-\Delta)$
end for
append $\textsc{Stop}$ to $\mathfrak{A}$; **return** start state and restoration $\mathfrak{A}$

kens, Reviser on edit-history tokens including MOVE tokens; since ∼20% of Reviser actions are moves, it sees fewer word tokens under this accounting). We report (i) with-input pairwise arena win rates judged by Skywork-Critic-8B (Skywork AI Alignment Team, 2024) with randomized candidate order, and (ii) evalPPL on continuation tokens under two evaluators—an AR scorer (GPT-2 Large; Radford et al., 2019) and a diffusion-style scorer (Dream-7B; Ye et al., 2025)—to reduce evaluator-family bias; we also report MAUVE (Pillutla et al., 2021) (shared 3k protocol, BERT pseudo-loglikelihood features, 100-token truncation). We do not report teacher-forced PPL over Reviser actions, since restoration trajectories are randomized and non-unique for a given output.

Reviser vs. AR and diffusion baselines. Against a same-architecture AR baseline trained on the same data, the judge prefers Reviser at both scales (Table 4: 61.30% at 100M, 54.40% at 300M)—a challenging setting, since the AR model is on its home data and optimizes exactly what evalPPL measures. Reviser also wins all three diffusion matchups by large margins (up to 85.93% vs. SEDD Small). Against publicly available size-matched AR models (Table 5), Reviser is preferred to all four at 100M and is competitive at 300M: it strongly beats Cerebras-GPT-256M (Dey et al., 2023) and is roughly tied with Pythia-410M (Biderman et al., 2023), GPT-2 Medium (Radford et al., 2019), and OPT-350M (Zhang et al., 2022).

Obfuscation (target → blank)	Restoration (blank → target)
1. Initial [Hi,|how are you?]	1. Initial [|]
2. DELETE [Hi|how are you?]	2. INSERT(how) [how|]
3. DELETE [|how are you?]	3. INSERT(are) [how are|]
4. MOVE(+4) [how are you?|]	4. INSERT(you) [how are you|]
5. DELETE [how are you|]	5. INSERT(?) [how are you?|]
6. DELETE [how are|]	6. MOVE(−4) [|how are you?]
7. DELETE [how|]	7. INSERT(Hi) [Hi|how are you?]
8. DELETE [|]	8. INSERT(,) [Hi,|how are you?]
	9. STOP [Hi,|how are you?]

Table 3: Side-by-side obfuscation and restoration trajectory for a full continuation without a prefix. Restoration reverses the obfuscation and inverts each step (DELETE ↔ INSERT of the deleted token, MOVE(Δ) ↔ MOVE($-\Delta$)).

Quality metrics. Table 6 shows Reviser has higher evalPPL than the AR baseline but far lower than SEDD/MDLM under both evaluators, and the strongest MAUVE in the comparison (Reviser 300M, 0.93; the AR baselines reach 0.90/0.91). The arena/evalPPL gap for Reviser is consistent with metric mismatch across generation paradigms: evalPPL uses an AR scorer while Reviser decodes via edit actions, and the very high SEDD/MDLM evalPPL reflects the same effect.

Trajectory statistics. Table 7 shows Reviser uses the edit interface in a strongly non-AR way: moves are frequent (28–33%), most moves are backward (∼65%), the mean insertion relative position is ∼0.41, only ∼3% of insertions are end-appends, and every evaluated example contains at least one backward revision—so the model is not merely emulating an end-append decoder. Move displacements concentrate on small jumps: at 100M, $|\Delta| \in \{1, 2, 4, 8\}$ account for 40.5/23.3/15.8/10.0% of moves (similar at 300M), consistent with frequent local revision punctuated by occasional long jumps.

Discussion. Two points are worth noting. First, the comparison is conservative: under the matched 30B-token budget, ∼20% of Reviser's training actions are MOVE tokens, so it is exposed to fewer word tokens than the AR baselines yet remains competitive. Second, the strong arena results despite higher evalPPL—together with the very high evalPPL of the SEDD/MDLM diffusion models, which are also non-AR generators—suggest that AR-scorer evalPPL systematically penalizes non-AR generation paradigms; arena win rates and MAUVE are therefore more faithful cross-paradigm quality signals in this setting.

Comparison (Reviser vs.)	Rev. WR ↑	Base. WR ↑
AR Baseline 100M	61.30%	38.70%
AR Baseline 300M	54.40%	45.60%
SEDD Small 169M	85.93%	14.07%
SEDD Absorb 353M	68.47%	31.53%
MDLM 170M	78.33%	21.67%

Table 4: Arena win rates (Skywork-Critic-8B judge) for Reviser vs. its size-matched AR baseline and vs. diffusion baselines. MDLM has no comparable 300M checkpoint in our setup.

Model	Size	Rev. WR ↑	AR WR ↑
Cerebras	111M	72.10%	27.90%
Cerebras	256M	71.30%	28.70%
Pythia	160M	68.80%	31.20%
Pythia	410M	51.30%	48.70%
GPT-2	117M	58.60%	41.40%
GPT-2	345M	50.40%	49.60%
OPT	125M	53.90%	46.10%
OPT	350M	49.90%	50.10%

Table 5: Reviser vs. publicly available size-matched AR baselines: arena win rates (Skywork-Critic-8B).

9 Conclusion

Reviser shows that a simple autoregressive-over-edits decoder—seeded with a deterministic prefix-seeding history and then decoding

Model	evalPPL ↓ GPT-2 L	evalPPL ↓ Dream-7B	MAUVE ↑
AR Baseline 100M	13.52	13.96	0.90
Reviser 100M	30.64	22.67	0.84
SEDD Small 169M	158.72	117.05	0.18
MDLM 170M	144.69	94.30	0.15
AR Baseline 300M	12.51	12.02	0.91
Reviser 300M	17.77	15.77	0.93
SEDD Absorb 353M	98.68	83.28	0.36

Table 6: Quality on C4 continuation: evalPPL (lower better) under an AR scorer (GPT-2 Large) and a diffusion scorer (Dream-7B), and MAUVE (higher better; shared 3k protocol).

Statistic (averaged across seeds)	100M	300M
Actions per output token	1.51	1.40
Insert fraction	67%	72%
Move fraction	33%	28%
Mean $\lvert\Delta\rvert$ move distance	4.92	6.05
Fraction backward moves	65.50%	64.50%
Mean insertion rel. position	0.41	0.42
Fraction end-appends	3.90%	3.10%
Examples with ≥ 1 backward rev.	100%	100%

Table 7: Reviser trajectory diagnostics in decoding. "Backward revision" means at least one backward move followed by a subsequent insertion into an earlier canvas region.

continuations via cursor actions—can produce genuinely non-monotonic text generation with a lightweight next-action interface. Across our experiments it performs frequent backward moves and mid-canvas insertions, is strongly preferred to the diffusion baselines we test, is competitive with roughly size-matched AR baselines, and stays close to AR in inference compute. Promising directions include richer edit operators (Delete/Replace/span edits), stronger state conditioning beyond the history-only trunk, training on real edit traces (or via DAgger/RL) rather than synthetic restoration trajectories, and structured-editing or agent applications where localized cursor edits may be more efficient than emitting full diffs. We view Reviser as a proof of concept that edit-history autoregression is a practical, scalable path to models that actively revise and improve their responses during generation.

10 Limitations

Reviser is sequential in action space, so decoding is not fully parallelizable over final-token positions; throughput remains constrained by step-by-step generation even though the model can revise non-monotonically. Quality also depends on the supervision trajectories: training on synthetic obfuscation–restoration paths does not reflect humanlike editing, which is typically purposeful and context-dependent rather than random, so the learned policy may differ from one trained on naturally occurring edits, and performance can degrade under distribution shift, with compounding error when the model visits states weakly represented in training data. The primary insert+move action set supports flexible revision, but some edits may require longer trajectories than richer operators (delete/replace/span). Finally, the history-only trunk must maintain an implicit representation of the current canvas from the edit-history stream; we did not observe this as a practical bottleneck at 100M/300M, and explicit state-conditioning (pooled canvas embeddings or cross-attention; Appendices A.2 and A.3) is available if needed. We expect that training on datasets of real edits—document revision histories or code-editing traces—could yield more efficient and semantically meaningful editing strategies.

11 Ethical Considerations

The Reviser models in this work are small research prototypes (100M and 300M parameters) trained on English web text and evaluated only on prefix continuation. They are not instruction-tuned, aligned, or safety-filtered, and our evaluation targets generation quality rather than factuality, bias, or safety; their outputs can reproduce the inaccuracies and social biases of web-scale corpora. We accordingly intend the checkpoints for research on the architecture rather than for deployment in user-facing systems.

A further consideration is specific to what Reviser does. Because it generates by applying localized insert and move edits to a mutable canvas, it is naturally a model for revising existing text rather than only producing new text—the running example in Section 1, where a backward move and a short insertion change the meaning of an already-written sentence, is exactly this kind of operation. That capability is dual-use: a revision-capable model can be steered to make small, targeted changes that

shift the meaning or attribution of a passage, to paraphrase text in ways that obscure its origin, or—using the richer edit operators we leave to future work—to rewrite content so as to evade authorship or machine-text detection. We flag this because it follows directly from the edit-based interface that is the core of our contribution, and we encourage work building on revision-capable generation to treat edit provenance and attribution as first-class concerns.

References

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*.

Stella Biderman, Hailey Schoelkopf, Quentin 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. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, and 11 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*.

Nolan Dey, Gurpreet Gosal, Zhiming Khachane, William Marshall, Ribhu Pathria, Marvin Tom, and Joel Hestness. 2023. Cerebras-GPT: Open compute-optimal language models trained on the Pile. Technical report, Cerebras Systems. ArXiv preprint arXiv:2304.03208.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. 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)*.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Jiatao Gu, Changhan Wang, and Jake Zhao. 2019. Levenshtein transformer. In *Advances in Neural Information Processing Systems*.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Conference on Empirical Methods in Natural Language Processing*.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy S. Liang, and Tatsunori B. Hashimoto. 2022. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*.

Aaron Lou, Chenlin Meng, and Stefano Ermon. 2024. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning*.

Xuezhe Ma, Chunting Zhou, Xian Li, Graham Neubig, and Eduard Hovy. 2019. Flowseq: Non-autoregressive conditional sequence generation with generative flow. 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)*.

Jonathan Mallinson, Aliaksei Severyn, Eric Malmi, and Guillermo Garrido. 2020. Felix: Flexible text editing through tagging and insertion. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. 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)*.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. MAUVE: Measuring the gap between neural text and human text using divergence frontiers. In *Advances in Neural Information Processing Systems*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI technical report*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*.

Subham Sekhar Sahoo, Marianne Arriola, Aaron Gokaslan, Edgar Mariano Marroquin, Alexander M. Rush, Yair Schiff, Justin T. Chiu, and Volodymyr Kuleshov. 2024. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*.

Skywork AI Alignment Team. 2024. Skywork-Critic-Llama-3.1-8B. Hugging Face model card, https://huggingface.co/Skywork/Skywork-Critic-Llama-3.1-8B.

Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. 2019. Insertion transformer: Flexible sequence generation via insertion operations. In *International Conference on Machine Learning*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. 2025. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*.

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. 2022. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

A Additional Reviser Variants

A.1 Other Edit Operators

This appendix subsection lists additional edit operators that can be supported by the same deterministic executor and validity masking used for INSERT/MOVE/STOP. More generally, any editing idea that can be represented by a token can be implemented as an action token (or a short composition of action tokens) and emitted by the transformer, provided the executor semantics are defined and validity-masked.

Replace the token immediately to the left of the cursor boundary with a new token $x \in V_A$:

$$\textsc{Replace}(x): \quad C_{t+1} \;=\; (c_{t,1}, \ldots, c_{t,u_t-1}, x, c_{t,u_t+1}, \ldots, c_{t,\ell_t}), \qquad u_{t+1} = u_t.$$

Validity constraint: $u_t \in \{1, \ldots, \ell_t\}$. (Equivalently, $\textsc{Replace}(x)$ can be viewed as DELETE followed by $\textsc{Insert}(x)$ at boundary $u_t - 1$, but we include it as a single macro-action.)

For a span length $k \in \{1, \ldots, u_t\}$, delete the last k tokens immediately to the left of the cursor boundary:

$$\textsc{SpanDelete}(k): \quad C_{t+1} \;=\; (c_{t,1}, \ldots, c_{t,u_t-k}, c_{t,u_t+1}, \ldots, c_{t,\ell_t}), \qquad u_{t+1} = u_t - k.$$

Validity constraint: $1 \le k \le u_t$.

Swap the last two tokens immediately to the left of the cursor boundary:

$$\textsc{Swap}: \quad C_{t+1} \;=\; (c_{t,1}, \ldots, c_{t,u_t-2}, c_{t,u_t}, c_{t,u_t-1}, c_{t,u_t+1}, \ldots, c_{t,\ell_t}), \qquad u_{t+1} = u_t.$$

Validity constraint: $u_t \in \{2, \ldots, \ell_t\}$.

Move a contiguous span of length k that ends at the token immediately left of the cursor boundary (i.e., indices $u_t - k + 1, \ldots, u_t$) to a destination boundary j in the pre-removal canvas:

$$\textsc{SpanMove}(k, j): \quad C_{t+1} =$$

$$\begin{cases} (\underbrace{c_{t,1}, \ldots, c_{t,j}}_{\text{left context}}, \underbrace{c_{t,u_t-k+1}, \ldots, c_{t,u_t}}_{\text{moved span}}, \underbrace{c_{t,j+1}, \ldots, c_{t,u_t-k}}_{\text{boosted context}}, \underbrace{c_{t,u_t+1}, \ldots, c_{t,\ell_t}}_{\text{right context}}), & \text{if } 1 \le j \le u_t - k, \\ (\underbrace{c_{t,1}, \ldots, c_{t,u_t-k}}_{\text{left context}}, \underbrace{c_{t,u_t+1}, \ldots, c_{t,j}}_{\text{collapsed context}}, \underbrace{c_{t,u_t-k+1}, \ldots, c_{t,u_t}}_{\text{moved span}}, \underbrace{c_{t,j+1}, \ldots, c_{t,\ell_t}}_{\text{right context}}), & \text{if } u_t \le j \le \ell_t. \end{cases}$$

$$u_{t+1} = \begin{cases} u_t, & \text{if } 1 \le j \le u_t - k, \\ j, & \text{if } u_t \le j \le \ell_t. \end{cases}$$

Validity constraints: $1 \le k \le u_t$ and $j \in \{1, \ldots, u_t - k\} \cup \{u_t, \ldots, \ell_t\}$ (enforced by masking). Underbraced segments may be empty.

Copy (without removing) the length-k span immediately left of the cursor boundary and insert it at boundary $j \in \{0, \ldots, \ell_t\}$:

$$\textsc{SpanCopy}(k, j): \quad C_{t+1} \;=\; (\underbrace{c_{t,1}, \ldots, c_{t,j}}_{\text{left context}}, \underbrace{c_{t,u_t-k+1}, \ldots, c_{t,u_t}}_{\text{copied context}}, \underbrace{c_{t,j+1}, \ldots, c_{t,\ell_t}}_{\text{right context}}), \qquad u_{t+1} = j + k,$$

with validity constraint $1 \le k \le u_t$. For both SPANCOPY and SPANMOVE, one can analogously copy/move spans taken from the right side of the cursor; we omit that symmetric math here.

All of the operators above admit straightforward supervised training using obfuscation restoration trajectories. The key observation is that each edit operator has a natural inverse. When temporary obfuscation-only insertions are used and later deleted, those transient insertions are treated as latent noise and excluded from the training loss because they are often random artifacts rather than meaningful target-response content.

A.2 Conditioning on the Canvas Representations via Pooled Canvas Embeddings

This subsection describes a variant in which action selection uses an explicit pooled summary of the current canvas. The motivation is simple: in a history-only trunk, canvas content is visible only indirectly through past actions and masking; adding a pooled canvas summary provides a direct channel for the model to "see" the current canvas state when choosing the next action. Our reported experiments use a history-only trunk without this module.

Attention pooling is just one instantiation of a broader idea: collapse variable-length canvas information into a fixed-size per-step vector (or small set of vectors) that can condition next-action prediction. There are many ways to perform this collapse (e.g., learned pooling queries, mean/max pooling with projections, convolutional pooling, recurrent summarizers, sparse/selective pooling, or learned routers). We use attention pooling here as a simple reference design.

Embed canvas tokens with the same shared token embedding table E as action tokens and a dedicated canvas positional table $P^{(C)}$. For the current canvas $C_t = (c_{t,1}, \ldots, c_{t,\ell_t})$, define

$$z_i \;=\; E[c_{t,i}] + P^{(C)}[i], \qquad i = 1, \ldots, \ell_t.$$

Let $q \in \mathbb{R}^d$ be a learned query vector. Compute attention weights:

$$s_i \;=\; q^\top z_i, \qquad \alpha_i \;=\; \frac{\exp(s_i)}{\sum_{j=1}^{\ell_t} \exp(s_j)}.$$

The pooled canvas embedding is:

$$\mu_t \;=\; \sum_{i=1}^{\ell_t} \alpha_i z_i \;\in\; \mathbb{R}^d.$$

In this variant, one injects the pooled canvas summary as an additive conditioning term on history-token inputs:

$$x_\tau \;=\; e_\tau \;+\; \mu_t, \quad \tau = 1, \ldots, t-1.$$

One may instead apply a learned linear projection $W_m \in \mathbb{R}^{d \times d}$:

$$x_\tau \;=\; e_\tau \;+\; W_m\, \mu_t,$$

including the identity (no projection) as the simplest choice.

A.3 Cross-Attention to Canvas Representations

Another optional design lets the history stream cross-attend to the full canvas sequence (often paired with or used instead of a single pooled canvas vector; Appendix A.2). Let $H_t \in \mathbb{R}^{m \times d}$ denote the history-stream hidden states at step t (the model's input token stream), and let $\mu_t \in \mathbb{R}^d$ be the pooled canvas embedding from Appendix A.2. Define $\mathbf{M}_t \in \mathbb{R}^{n \times d}$ as the pooled-canvas matrix whose rows are all equal to $\mu_t^\top$. In a cross-attention block, we compute queries from the history and keys/values from this pooled-canvas matrix:

$$Q_t = H_t W_Q, \qquad K_t = \mathbf{M}_t W_K, \qquad V_t = \mathbf{M}_t W_V,$$

and add the resulting cross-attention output as a residual update:

$$\widetilde{H}_t = H_t + \textsc{Softmax}\Big(\frac{Q_t K_t^\top}{\sqrt{d_k}} + M\Big) V_t.$$

In our main implementation, cross-attention is inserted after the history self-attention (and before the MLP) in each transformer layer, though in practice it often suffices to apply it only in the last L_{ca} layers to reduce cost. The cross-attention update can also be gated:

$$\widetilde{H}_t = H_t + \lambda_{\text{ca}} \cdot \textsc{Softmax}\Big(\frac{Q_t K_t^\top}{\sqrt{d_k}} + M\Big) V_t,$$

with either a learned scalar λ_{ca} per layer or a small gating MLP. With this pooled-canvas matrix, the block uses a compressed canvas memory derived from μ_t rather than per-token canvas states; in practice one can set $n = 1$ for minimal cost or use small n for implementation convenience.

A.4 DAgger with an Alignment-Based Oracle

Supervised training on fixed restoration trajectories can lead to distribution shift at inference time. DAgger (Ross et al., 2011) mitigates this by iterating between rolling out the current policy and retraining on oracle-labeled states from those rollouts.

Our oracle aligns the current canvas C_t against the target $C^\star$ via LCS-style sequence alignment, labeling tokens as matched (anchors) or mismatched. From this alignment it constructs a deterministic edit script: when the cursor is in a matched region, the oracle emits MOVE actions to route to mismatched regions; when the cursor is in a mismatched region, the oracle emits local content edits (e.g., INSERT and, when available, DELETE). DAgger then aggregates oracle-labeled states into the training set and re-optimizes by teacher forcing:

$$\min_\theta \; \mathbb{E}_{(s,a^\star)\sim\mathcal{D}}\big[-\log p_\theta(a^\star \mid s)\big].$$

A mixture policy that occasionally defers to the oracle prevents catastrophic divergence during early iterations, with the oracle mixing probability decayed over training so the policy gradually assumes full control.

A.5 Reinforcement Learning Fine-Tuning

While supervised restoration provides strong local imitation signals, it does not directly optimize sequence-level objectives such as holistic response quality, preference alignment, or length/compute trade-offs. RL fine-tuning is a natural extension for optimizing such non-decomposable rewards over complete edit trajectories.

Editing is formalized as an episodic MDP: states s_t encode (C_t, u_t, H_t) plus a validity mask; actions are the edit operators; transitions are deterministic via the executor. Terminal rewards come from an LLM judge that ranks K candidate outputs and assigns a linearly decayed preference reward $R_{\text{pref}}(A^{(\pi(r))}) = 1 - (r-1)/(K-1)$.

Policy optimization uses PPO-style clipping with entropy regularization, an optional KL penalty to the supervised reference policy, and an optional action-type regularizer:

$$\mathcal{L}(\theta,\phi) = -\mathcal{L}_{\text{policy}}(\theta) + c_v\,\mathcal{L}_{\text{value}}(\phi) - c_e\,\mathcal{L}_{\text{ent}}(\theta) + c_{\text{kl}}\,\mathcal{L}_{\text{kl}}(\theta) + c_{\text{type}}\,\mathcal{L}_{\text{type}}(\theta).$$

To reduce early-rollout collapse, rollouts are warm-started with a decaying prefix of oracle actions. Extensions to RLHF-style preference optimization or multi-objective length/compute penalties are straightforward within this MDP formulation.

B Prior Work Mechanisms and FLOPs-Based Efficiency Accounting

B.1 Goal and definitions (FLOPs-based, shared constants across models)

We quantify inefficiency by comparing inference FLOPs to produce a full length-n output under each method's decoding procedure, relative to an autoregressive (AR) baseline.

We fix the same backbone hyperparameters for every method (so ratios are meaningful):

$$n = 128, \quad |V_c| = 50{,}000, \quad d = d_{\text{model}} = 768, \quad L = 12, \quad d_{\text{ff}} = 4d, \quad K_{\max} = 16.$$

For Reviser, the action vocabulary size is $|V_a| = |V_c| + |\mathcal{M}| + 1$; since $|\mathcal{M}| \ll |V_c|$, we often use $|V_a| \approx |V_c|$ when it does not materially affect ratios.

We count dominant matrix-multiply compute and omit small terms (biases, layernorm, activations, elementwise ops). All FLOPs expressions in this appendix are built from primitives in Appendix B.2. We count one fused multiply-accumulate (MAC) as one unit in $F_{\text{mult}}(a, b, c) = abc$. (To convert to conventions where one MAC equals two FLOPs, multiply all reported $F(\cdot)$ values by 2; ratios are unchanged.)

For each method $\mathcal{M}$, define $F^{\mathcal{M}}_{\text{infer}}(n)$ as the total FLOPs required to output an entire length-n sequence under that method's inference procedure (summing all model calls, scoring/rounding steps, termination checks, etc.). We define inefficiency as $I^{\mathcal{M}}_{\text{infer}} \triangleq \frac{F^{\mathcal{M}}_{\text{infer}}(n)}{F^{\text{AR}}_{\text{infer}}(n)}$.

B.2 Primitive FLOPs functions

Let $A \in \mathbb{R}^{a\times b}$ and $B \in \mathbb{R}^{b\times c}$. We define $F_{\text{mult}}(a, b, c) \triangleq abc$, i.e., one multiply-accumulate (MAC) is one unit of compute.

We count dominant matrix-multiply compute and omit small terms (biases, layernorm, activations, elementwise ops). Here L denotes Transformer depth (number of layers), while n denotes sequence length. Because attention structure differs across settings, we distinguish full (bidirectional) attention from causal (lower-triangular) attention.

Define $F^{\text{full}}_{\text{transformer}}(n)$ as the dominant FLOPs for one forward pass through an L-block Transformer on a length-n sequence with full self-attention:

$$F^{\text{full}}_{\text{transformer}}(n) \triangleq L\Big(4\,F_{\text{mult}}(n, d, d) \;+\; 2\,F_{\text{mult}}(n, n, d) \;+\; 2\,F_{\text{mult}}(n, d, d_{\text{ff}})\Big).$$

With $d_{\text{ff}} = 4d$:

$$F^{\text{full}}_{\text{transformer}}(n) = L\left(12nd^2 + 2n^2d\right).$$

Causal self-attention uses only the $n(n+1)/2$ lower-triangular query–key pairs. We therefore model the attention-score and attention-apply matmuls by replacing the full n^2 pair count with $n(n+1)/2$. Define the dominant FLOPs for one forward pass with causal self-attention as

$$F^{\text{causal}}_{\text{transformer}}(n) \triangleq L\Big(4\,F_{\text{mult}}(n, d, d) \;+\; n(n+1)\,d \;+\; 2\,F_{\text{mult}}(n, d, d_{\text{ff}})\Big),$$

where $n(n+1)d$ comes from summing the triangular costs of $QK^\top$ and $\text{Attn}\cdot V$, each equal to $\frac{n(n+1)}{2}d$. With $d_{\text{ff}} = 4d$:

$$F^{\text{causal}}_{\text{transformer}}(n) = L\left(12nd^2 + n^2d + nd\right).$$

Under this triangular convention, a causal forward pass at length n (computed in parallel under a causal mask) and KV-cached autoregressive decoding up to token n have the same dominant attention-matmul count.

Encoder self-attention is bidirectional, so we use the full-attention expression $F_{\text{enc}}(n) \triangleq F^{\text{full}}_{\text{transformer}}(n)$.

We model one decoder cross-attention layer (dominant matmuls) between a target length-n sequence and a source length-n_{src} sequence as:

$$\begin{aligned} F_{\text{xattn,layer}}(n, n_{\text{src}}) \triangleq\; & \underbrace{F_{\text{mult}}(n, d, d)}_{\text{Q proj (tgt)}} + \underbrace{2F_{\text{mult}}(n_{\text{src}}, d, d)}_{\text{K,V proj (src)}} + \underbrace{F_{\text{mult}}(n, d, d)}_{\text{O proj (tgt)}} \\ & + \underbrace{F_{\text{mult}}(n, d, n_{\text{src}})}_{\text{attn logits } (QK^\top)} + \underbrace{F_{\text{mult}}(n, n_{\text{src}}, d)}_{\text{attn apply } (\text{Attn}\cdot V)}. \end{aligned}$$

Stacking across L decoder layers gives $F_{\text{xattn}}(n, n_{\text{src}}) \triangleq L\,F_{\text{xattn,layer}}(n, n_{\text{src}})$.

We model a dense vocab projection over m positions as $F_{\text{vocab}}(m) \triangleq F_{\text{mult}}(m, d, |V_c|)$.

C Model-by-model FLOPs accounting

C.1 Group 1. Anchors

C.1.1 Autoregressive Transformer (AR baseline)

We model the total dominant matmul compute of KV-cached AR decoding up to length n using the causal (triangular) attention convention. Across n decoding steps, the vocab head is applied once per generated token (total n applications), yielding:

$$F_{\text{infer}}^{\text{AR}}(n) \triangleq F_{\text{transformer}}^{\text{causal}}(n) + F_{\text{vocab}}(n).$$

Using $n = 128$, $|V_c| = 50{,}000$, $d = 768$, $L = 12$, and $d_{\text{ff}} = 4d$:

$$F_{\text{transformer}}^{\text{causal}}(128) \approx 11.02\,\text{G}, \qquad F_{\text{vocab}}(128) \approx 4.92\,\text{G},$$

so

$$F_{\text{infer}}^{\text{AR}}(128) \approx 15.94\,\text{G}.$$

By definition, $I_{\text{infer}}^{\text{AR}} = 1$.

C.1.2 Cursor-style Edit Generation: Reviser (this work)

Reviser treats text generation as an editing process over a cursor position. At each step it predicts exactly one action (e.g., INSERT(token), or MOVE(Δ)), applies that edit to the current sequence, and continues iterating until termination. Unlike prior edit transformers that emit large per-position edit heads, Reviser emits a single action per step with a normal-sized token vocabulary head (plus a small set of edit actions), and it is trained with standard supervised learning (teacher forcing) in the same way as a typical Transformer.

Let p_{move} be the fraction of actions that are cursor-moves. To produce n inserted tokens, the expected number of action steps is $n_{\text{eff}} = \frac{n}{1-p_{\text{move}}}$. For $n = 128$, we report three settings: $p_{\text{move}} \in \{0.20, 0.28, 0.33\}$, giving $n_{\text{eff}} \in \{160,\ 177.78,\ 191.04\}$.

We approximate Reviser's total decoding compute by evaluating the AR causal proxy at the effective length:

$$F_{\text{infer}}^{\text{Reviser}}(n) \triangleq F_{\text{transformer}}^{\text{causal}}(n_{\text{eff}}) + F_{\text{vocab}}(n_{\text{eff}}).$$

$$I_{\text{infer}}^{\text{Reviser}}(n) \triangleq \frac{F_{\text{infer}}^{\text{Reviser}}(n)}{F_{\text{infer}}^{\text{AR}}(n)}.$$

Using $F_{\text{infer}}^{\text{AR}}(128) \approx 15.94\,\text{G}$, we obtain:

$$\begin{aligned}
p_{\text{move}} = 0.20: \quad & F_{\text{transformer}}^{\text{causal}}(160) \approx 13.83\,\text{G},\ F_{\text{vocab}}(160) \approx 6.14\,\text{G}, \\
& F_{\text{infer}}^{\text{Reviser}}(128) \approx 19.97\,\text{G},\ I_{\text{infer}}^{\text{Reviser}} \approx 1.25. \\
p_{\text{move}} = 0.28: \quad & F_{\text{transformer}}^{\text{causal}}(177.78) \approx 15.39\,\text{G},\ F_{\text{vocab}}(177.78) \approx 6.83\,\text{G}, \\
& F_{\text{infer}}^{\text{Reviser}}(128) \approx 22.22\,\text{G},\ I_{\text{infer}}^{\text{Reviser}} \approx 1.39. \\
p_{\text{move}} = 0.33: \quad & F_{\text{transformer}}^{\text{causal}}(191.04) \approx 16.56\,\text{G},\ F_{\text{vocab}}(191.04) \approx 7.34\,\text{G}, \\
& F_{\text{infer}}^{\text{Reviser}}(128) \approx 23.90\,\text{G},\ I_{\text{infer}}^{\text{Reviser}} \approx 1.50.
\end{aligned}$$

C.2 Group 2. Diffusion NAR LMs (100M)

Many diffusion-style language models perform generation as T_{dec} repeated full-sequence Transformer forward passes, each followed by a vocabulary projection. For methods with this structure, we define the generation FLOPs as

$$F_{\text{diffusion}}^{\mathcal{M}_{\text{diff}}}(n, T_{\text{dec}}) \triangleq T_{\text{dec}} \left(F_{\text{transformer}}^{\text{full}}(n) + F_{\text{vocab}}(n) \right).$$

C.2.1 SEDD

Lou et al. (2024) parameterize SEDD, a reverse discrete diffusion process, and generate by running a fixed number T_{dec} of "network evaluations" (function evaluations) of Transformer conditioned on time and noise.

A conservative matmul proxy is that each step computes full-sequence hidden states (full attention) and produces categorical scores for all n positions:

$$F_{\text{infer}}^{\text{SEDD}}(n, T_{\text{dec}}) \triangleq F_{\text{diffusion}}^{\text{SEDD}}(n, T_{\text{dec}})$$

We report two representative step counts:

$$T_{\text{dec}} = 32: \quad F_{\text{infer}}^{\text{SEDD}}(128, 32) \approx 514.84\,\text{G}, \quad I_{\text{infer}}^{\text{SEDD}} = 32.30.$$

$$T_{\text{dec}} = 2048: \quad F_{\text{infer}}^{\text{SEDD}}(128, 2048) \approx 32{,}949.92\,\text{G}, \quad I_{\text{infer}}^{\text{SEDD}} = 2{,}067.25.$$

C.2.2 MDLM

Sahoo et al. (2024) introduce MDLM, a masked discrete diffusion language model. Sampling starts from an all-`[MASK]` sequence and runs a discretized reverse diffusion with T_{dec} steps; at each step, the model predicts token distributions conditioned on the current partially denoised sequence.

Using the same conservative "full logits per step" proxy:

$$F_{\text{infer}}^{\text{MDLM}}(n, T_{\text{dec}}) \triangleq F_{\text{diffusion}}^{\text{MDLM}}(n, T_{\text{dec}})$$

$$F_{\text{infer}}^{\text{MDLM}}(128, 1000) \approx 16{,}088.83\,\text{G}, \quad I_{\text{infer}}^{\text{MDLM}} = 1{,}009.40.$$

C.2.3 D3PM

Austin et al. (2021) introduce D3PM, a general discrete denoising diffusion probabilistic model. In the text setting, it corresponds to a time-conditioned denoiser run for T_{dec} discrete reverse steps to transform a highly corrupted sequence into a clean sample.

$$F_{\text{infer}}^{\text{D3PM}}(n, T_{\text{dec}}) \triangleq F_{\text{diffusion}}^{\text{D3PM}}(n, T_{\text{dec}})$$

$$F_{\text{infer}}^{\text{D3PM}}(128, 1000) \approx 16{,}088.83\,\text{G}, \quad I_{\text{infer}}^{\text{D3PM}} = 1{,}009.40.$$

C.2.4 Diffusion-LM

Li et al. (2022) propose Diffusion-LM, which performs continuous diffusion over a length-n sequence of vectors and runs a (bidirectional) Transformer denoiser for T_{dec} reverse steps. To convert continuous vectors to discrete tokens, we upper-bound discretization/rounding by a dense vocab scoring matmul.

Let $|V|$ denote the (discrete) output vocabulary size for this model. We define

$$F_{\text{round}}(n) \triangleq F_{\text{mult}}(n, d, |V|),$$

(which equals $F_{\text{vocab}}(n)$ under our shared 100M configuration where $|V| = |V_c|$).

Each reverse step runs the denoiser once over the full length-n sequence (full attention):

$$F_{\text{denoise}}(n) \triangleq F_{\text{transformer}}^{\text{full}}(n).$$

The clamping trick optionally applies rounding to the predicted x_0 on a subset of the non-final reverse steps. Let $\gamma_{\text{clamp}} \in [0, 1]$ denote the fraction of the first $T_{\text{dec}} - 1$ steps on which clamping (rounding) is applied.

Method	Variant	$F^{\mathcal{M}}_{\text{infer}}$ (G)	$I^{\mathcal{M}}_{\text{infer}}$
AR (100M)	baseline	15.94	1
Reviser	$p_{\text{move}} \in \{0.20, 0.28, 0.33\}$	19.97–23.90	1.25–1.50
SEDD	T_{dec}=32 (full logits each step)	514.84	32.30
SEDD	T_{dec}=2048 (full logits each step)	32,949.92	2,067.25
MDLM	T_{dec}=1000 (ancestral; full logits each step)	16,088.83	1,009.40
D3PM	T_{dec}=1000 (discrete diffusion; full logits each step)	16,088.83	1,009.40
Diffusion-LM	T_{dec}=200, γ_{clamp}=0	2,239.64	140.51
Diffusion-LM	T_{dec}=200, γ_{clamp}=1	3,217.77	201.88
Diffusion-LM	T_{dec}=2000, γ_{clamp}=0	22,352.17	1,402.36

Table 8: 100M-tier diffusion-family FLOPs multipliers under a matmul-dominant proxy. (All rows use the shared 100M reference configuration.)

We pay T_{dec} denoiser evaluations, plus rounding once at the end, plus clamping-rounding on $\gamma_{\text{clamp}}(T_{\text{dec}} - 1)$ non-final steps:

$$F^{\text{DiffLM}}_{\text{infer}}(n) \triangleq T_{\text{dec}}\, F_{\text{denoise}}(n) + \Big(1 + \gamma_{\text{clamp}}(T_{\text{dec}} - 1)\Big)\, F_{\text{round}}(n).$$

Using $F^{\text{full}}_{\text{transformer}}(128) \approx 11.17\,\text{G}$ and $F_{\text{round}}(128) = F_{\text{mult}}(128, 768, 50000) \approx 4.92\,\text{G}$, we obtain:

$$T_{\text{dec}} = 200,\ \gamma_{\text{clamp}} = 0: \quad F^{\text{DiffLM}}_{\text{infer}}(128) \approx 2{,}239.64\,\text{G}, \quad I^{\text{DiffLM}}_{\text{infer}} = 140.51.$$

$$T_{\text{dec}} = 200,\ \gamma_{\text{clamp}} = 1: \quad F^{\text{DiffLM}}_{\text{infer}}(128) \approx 3{,}217.77\,\text{G}, \quad I^{\text{DiffLM}}_{\text{infer}} = 201.88.$$

$$T_{\text{dec}} = 2000,\ \gamma_{\text{clamp}} = 0: \quad F^{\text{DiffLM}}_{\text{infer}}(128) \approx 22{,}352.17\,\text{G}, \quad I^{\text{DiffLM}}_{\text{infer}} = 1{,}402.36.$$

C.2.5 Summary table (100M tier)

C.3 Group 3: Large-Scale Diffusion NAR LMs

The large backbones used in this subsection (e.g., LLaMA/Qwen families) commonly use (i) a gated MLP with three linear maps (`gate_proj`, `up_proj`, `down_proj`), and (ii) grouped-query attention (GQA) where K/V have fewer heads than Q. Let h be the number of query heads and h_{kv} the number of K/V heads. With head dimension $d_{\text{head}} = d/h$, define the effective K/V projection width $d_{\text{kv}} \triangleq h_{\text{kv}}\, d_{\text{head}}$. (For standard MHA, $h_{\text{kv}} = h$ so $d_{\text{kv}} = d$.)

For n positions, Q and O are $(n, d) \times (d, d)$, while K and V are $(n, d) \times (d, d_{\text{kv}})$:

$$F^{\text{GQA}}_{\text{proj}}(n; d, d_{\text{kv}}) \triangleq 2F_{\text{mult}}(n, d, d) + 2F_{\text{mult}}(n, d, d_{\text{kv}}).$$

A gated MLP uses three matmuls (gate, up, down):

$$F^{\text{gated}}_{\text{mlp}}(n; d, d_{\text{ff}}) \triangleq 3F_{\text{mult}}(n, d, d_{\text{ff}}).$$

$$F^{\text{full,gated}}_{\text{transformer}}(n; d, L, d_{\text{ff}}, d_{\text{kv}}) \triangleq L\Big(F^{\text{GQA}}_{\text{proj}}(n; d, d_{\text{kv}}) + 2F_{\text{mult}}(n, n, d) + F^{\text{gated}}_{\text{mlp}}(n; d, d_{\text{ff}})\Big).$$

$$F^{\text{causal,gated}}_{\text{transformer}}(n; d, L, d_{\text{ff}}, d_{\text{kv}}) \triangleq L\Big(F^{\text{GQA}}_{\text{proj}}(n; d, d_{\text{kv}}) + n(n+1)d + F^{\text{gated}}_{\text{mlp}}(n; d, d_{\text{ff}})\Big).$$

C.3.1 Large-scale reference configurations (native sizes)

We keep each large model at its native architecture and compare to a size-matched AR baseline of the same scale. All comparisons use the same output length $n = 128$ and the same FLOPs conventions as in Appendix B.2.

For a decoder-only Transformer, we denote the number of layers by L, hidden size by d, MLP width by d_{ff}, and vocabulary size by $|V|$. For attention, let h be the number of query heads and h_{kv} the number of K/V heads (Grouped-Query Attention, GQA). With head dimension $d_{\text{head}} \triangleq d/h$, the effective K/V projection width is $d_{\text{kv}} \triangleq h_{\text{kv}}\, d_{\text{head}}$. (For standard multi-head attention, $h_{\text{kv}} = h$ so $d_{\text{kv}} = d$.)

$$L = 64, \quad d = 5120, \quad d_{\text{ff}} = 27648, \quad |V| = 152064, \quad h = 40, \quad h_{\text{kv}} = 8.$$

Here $d_{\text{head}} = 5120/40 = 128$ and $d_{\text{kv}} = 8 \cdot 128 = 1024$.

$$L = 28, \quad d = 3584, \quad d_{\text{ff}} = 18944, \quad |V| = 152064, \quad h = 28, \quad h_{\text{kv}} = 4.$$

Here $d_{\text{head}} = 3584/28 = 128$ and $d_{\text{kv}} = 4 \cdot 128 = 512$.

For LLaDA-8B we use the published LLaDA config (full-attention masked-token predictor):

$$\text{LLaDA-8B: } L = 32, \quad d = 4096, \quad d_{\text{ff}} = 12288, \quad |V| = 126464, \quad h = 32, \quad h_{\text{kv}} = 32,$$

so $d_{\text{head}} = 4096/32 = 128$ and $d_{\text{kv}} = 32 \cdot 128 = 4096$ (no GQA reduction).

For the size-matched AR baseline we use LLaMA3-8B (causal decoder with GQA):

$$\text{LLaMA3-8B (AR): } L = 32, \quad d = 4096, \quad d_{\text{ff}} = 14336, \quad |V| = 128256, \quad h = 32, \quad h_{\text{kv}} = 8,$$

so $d_{\text{head}} = 128$ and $d_{\text{kv}} = 8 \cdot 128 = 1024$.

C.3.2 AR baselines (size-matched)

For each backbone, we use a LLaMA/Qwen-style decoder proxy (GQA + gated MLP):

$$F_{\text{infer}}^{\text{AR}}(n) \triangleq F_{\text{transformer}}^{\text{causal,gated}}(n; d, L, d_{\text{ff}}, d_{\text{kv}}) + F_{\text{mult}}(n, d, |V|).$$

$$F_{\text{infer}}^{\text{AR (Qwen2.5-32B)}}(128) \approx 4{,}099.39\,\text{G}.$$

$$F_{\text{infer}}^{\text{AR (Qwen2.5-7B)}}(128) \approx 906.65\,\text{G}.$$

$$F_{\text{infer}}^{\text{AR (LLaMA3-8B)}}(128) \approx 962.76\,\text{G}.$$

C.3.3 SDLM-32B-D4 (Sequential Diffusion Language Model)

SDLM predicts a fixed block of D tokens per forward pass, then selects the longest high-confidence prefix (length $k \in \{1, \ldots, D\}$) and commits it. KV-cache states for committed tokens are reused. Here, D is the block size (we use $D = 4$), and $\bar{k} \triangleq \mathbb{E}[k]$ is the average committed tokens per pass.

To propose a block of D new tokens given a cached prefix of length n_{pre}, we count: (i) Q/K/V/O projections + gated MLP on the D new positions, and (ii) attention matmuls for D queries attending to the cached prefix plus the causal within-block keys. Let d_{kv} be the K/V projection width (for GQA; for MHA, $d_{\text{kv}} = d$). We use:

$$F_{\text{kvblk}}(D, n_{\text{pre}}; d_{\text{ff}}, d_{\text{kv}}) \triangleq L\Big(\underbrace{2F_{\text{mult}}(D, d, d) + 2F_{\text{mult}}(D, d, d_{\text{kv}})}_{\text{Q,O and K,V projections}} + \underbrace{3F_{\text{mult}}(D, d, d_{\text{ff}})}_{\text{gated MLP}} + \underbrace{2F_{\text{mult}}(D, n_{\text{pre}}, d) + F_{\text{mult}}(D, D{+}1, d)}_{\text{attn (prefix + causal within-block)}}\Big).$$

With $S \approx \lceil n/\bar{k} \rceil$ passes and prefix length $n_{\text{pre}} \approx (s-1)\bar{k}$:

$$F_{\text{infer}}^{\text{SDLM}}(n) \triangleq \sum_{s=1}^{S} \Big(F_{\text{kvblk}}(D, (s-1)\bar{k}; d_{\text{ff}}, d_{\text{kv}}) + F_{\text{mult}}(D, d, |V|) \Big).$$

$$F_{\text{infer}}^{\text{SDLM-32B-D4}}(128) \approx 8{,}198.94\,\text{G}, \quad I_{\text{infer}}^{\text{SDLM-32B-D4}} = 2.00.$$

C.3.4 Dream-7B (diffusion LLM; Qwen2.5-7B backbone)

Dream performs masked discrete diffusion: starting from a fully masked response, it iteratively denoises for T_{dec} steps using a full-attention Transformer denoiser.

Let m_t be the number of masked positions scored at step t. For a simple closed form we approximate a linear schedule: $\sum_{t=1}^{T_{\text{dec}}} m_t \approx n\frac{T_{\text{dec}}+1}{2}$.

$$F_{\text{infer}}^{\text{Dream}}(n) \approx T_{\text{dec}}\, F_{\text{transformer}}^{\text{full}}(n) + \frac{T_{\text{dec}}+1}{2}\, F_{\text{vocab}}(n; |V|).$$

$$T_{\text{dec}} = 10: \quad F_{\text{infer}}^{\text{Dream-7B}}(128) \approx 8{,}768.93\,\text{G}, \quad I_{\text{infer}}^{\text{Dream-7B}} = 9.67.$$
$$T_{\text{dec}} = 20: \quad F_{\text{infer}}^{\text{Dream-7B}}(128) \approx 17{,}502.98\,\text{G}, \quad I_{\text{infer}}^{\text{Dream-7B}} = 19.31.$$

C.3.5 LLaDA-8B (diffusion mask predictor; compared to LLaMA3-8B AR)

LLaDA predicts masked tokens with a full-attention Transformer (no causal mask) and runs a reverse masking/denoising process for T_{dec} steps. For instruct-style decoding, it can also generate left-to-right in blocks, applying the reverse process inside each block. Because this configuration uses $h_{\text{kv}}{=}h$ (no GQA reduction), its K/V-side projection terms are larger than in GQA AR baselines, which contributes to the larger compute multiplier.

$$F_{\text{infer}}^{\text{LLaDA}}(n) \approx T_{\text{dec}}\, F_{\text{transformer}}^{\text{full}}(n) + \frac{T_{\text{dec}}+1}{2}\, F_{\text{vocab}}(n; |V|).$$

Partition the response into $B = \lceil n/b \rceil$ blocks of size b, and run T_b reverse steps per block. We approximate full-attention trunk length as $n_j = \min(jb, n)$ for block j, and vocab logits only for the b block positions:

$$F_{\text{infer,blk}}^{\text{LLaDA}}(n) \triangleq \sum_{j=1}^{B} \Big(T_b\, F_{\text{transformer}}^{\text{full}}(n_j) + \frac{T_b+1}{2}\, F_{\text{vocab}}(b; |V|) \Big).$$

$$T_{\text{dec}} = 256: \quad F_{\text{infer}}^{\text{LLaDA-8B}}(128) \approx 238{,}317.94\,\text{G}, \quad I_{\text{infer}}^{\text{LLaDA-8B}} = 247.54.$$
$$\text{semi-AR: } b = 32,\ T_b = 16: \quad F_{\text{infer,blk}}^{\text{LLaDA-8B}}(128) \approx 36{,}426.56\,\text{G}, \quad I_{\text{infer,blk}}^{\text{LLaDA-8B}} = 37.84.$$

C.3.6 Summary table (large-scale)

C.4 Group 4. Refinement NAR LMs (MT)

C.4.1 Levenshtein Transformer (LevT)

Gu et al. (2019) describe the Levenshtein Transformer (LevT), which runs R refinement iterations. Each iteration applies three predictors in sequence: delete π^{del}, placeholder-count π^{plh}, and token-fill π^{tok}. This corresponds to three trunk evaluations per iteration, with optional early exit for the first two: $f_{\text{del}} = \frac{b_{\text{del}}}{L}$ and $f_{\text{plh}} = \frac{b_{\text{plh}}}{L}$, with $f_{\text{del}}, f_{\text{plh}} \in (0, 1]$.

We evaluate delete + placeholder predictors at an average length $n_0 \triangleq 0.8n$, and we evaluate the token-fill predictor at length n (after insertion returns to the target length). For the token-fill vocabulary head, we assume only placeholder positions require vocab logits, with $m \triangleq 0.4n$ denoting the estimated number of placeholder tokens to fill at each refinement iteration. This LevT accounting is a decoder-side iterative-edit proxy (matching the LevT edit cycle) rather than a full source-conditioned MT accounting with separate encoder/cross-attention terms.

Method	Variant	$F^{\mathcal{M}}_{\text{infer}}$ (G)	$I^{\mathcal{M}}_{\text{infer}}$
AR (Qwen2.5-32B)	baseline	4,099.39	1
SDLM-32B-D4	D=4, $\bar{k}$=2 (no verify)	8,198.94	2.00
AR (Qwen2.5-7B)	baseline	906.65	1
Dream-7B	T_{dec}=10 (lin. unmask)	8,768.93	9.67
Dream-7B	T_{dec}=20 (lin. unmask)	17,502.98	19.31
AR (LLaMA3-8B)	baseline	962.76	1
LLaDA-8B	T_{dec}=256 (full-length)	238,317.94	247.54
LLaDA-8B	semi-AR remask: b=32, T_b=16	36,426.56	37.84

Table 9: Large-scale diffusion-family FLOPs multipliers under a matmul-dominant proxy. Each block uses a size-matched AR baseline (Qwen2.5-32B, Qwen2.5-7B, or LLaMA3-8B).

LevT runs bidirectional/refinement-style passes, so we use full attention in the trunk terms:

$$\begin{aligned} F^{\text{LevT}}_{\text{iter}}(n) \;\approx\; & \underbrace{f_{\text{del}}\, F^{\text{full}}_{\text{transformer}}(n_0)}_{\text{delete trunk}} + \underbrace{f_{\text{plh}}\, F^{\text{full}}_{\text{transformer}}(n_0)}_{\text{placeholder trunk}} + \underbrace{F^{\text{full}}_{\text{transformer}}(n)}_{\text{token trunk}} \\ & + \underbrace{F_{\text{mult}}(n_0, d, 2)}_{\text{delete head}} + \underbrace{F_{\text{mult}}(n_0+1, 2d, K_{\max}+1)}_{\text{placeholder-count head}} + \underbrace{F_{\text{vocab}}(m)}_{\text{vocab projection on placeholders}}, \end{aligned}$$

where $n_0 = 0.8n$ and $m = 0.4n$. These are average-case analytical surrogates and need not be integers. Total decoding cost is

$$F^{\text{LevT}}_{\text{infer}}(n) \approx R \cdot F^{\text{LevT}}_{\text{iter}}(n).$$

Using the shared reference configuration with $n = 128$, $n_0 = 102.4$, and $m = 51.2$:

$$F^{\text{full}}_{\text{transformer}}(102.4) \approx 8.89\,\text{G}, \quad F^{\text{full}}_{\text{transformer}}(128) \approx 11.17\,\text{G},$$

and head costs

$$F_{\text{mult}}(102.4, 768, 2) \approx 0.16\,\text{M}, \quad F_{\text{mult}}(103.4, 1536, 17) \approx 2.70\,\text{M}, \quad F_{\text{vocab}}(51.2) \approx 1.97\,\text{G}.$$

Set $f_{\text{del}} = f_{\text{plh}} = 1$. Then:

$$F^{\text{LevT}}_{\text{iter,noEE}}(128) \approx 30.92\,\text{G}.$$

With $R = 5$ and $R = 10$:

$$F^{\text{LevT}}_{\text{infer,noEE}}(128, R=5) \approx 154.62\,\text{G}, \qquad F^{\text{LevT}}_{\text{infer,noEE}}(128, R=10) \approx 309.24\,\text{G}.$$

Set $f_{\text{del}} = f_{\text{plh}} = 0.5$. Then:

$$F^{\text{LevT}}_{\text{iter,EE}}(128) \approx 22.03\,\text{G}.$$

With $R = 5$ and $R = 10$:

$$F^{\text{LevT}}_{\text{infer,EE}}(128, R=5) \approx 110.17\,\text{G}, \qquad F^{\text{LevT}}_{\text{infer,EE}}(128, R=10) \approx 220.33\,\text{G}.$$

Using $F^{\text{AR}}_{\text{infer}}(128) \approx 15.94\,\text{G}$:

$$I^{\text{LevT}}_{\text{infer,noEE}}(R=5) = 9.70, \quad I^{\text{LevT}}_{\text{infer,noEE}}(R=10) = 19.40,$$

$$I^{\text{LevT}}_{\text{infer,EE}}(R=5) = 6.91, \quad I^{\text{LevT}}_{\text{infer,EE}}(R=10) = 13.82.$$

C.4.2 Insertion-based Generation: Insertion Transformer (Stern et al., 2019)

Stern et al. (2019) propose the Insertion Transformer (InsT), which generates a sequence by iteratively inserting tokens into slots (gaps). At iteration t, the current partial sequence has length n_t. The model runs a Transformer pass on the current sequence and predicts, for each slot, either a vocabulary token to insert or a special `no-insert` action.

Insertion Transformer uses bidirectional passes over the current partial sequence, so we use full attention:

$$F_{\text{iter}}^{\text{InsT}}(n_t) \triangleq F_{\text{transformer}}^{\text{full}}(n_t) + F_{\text{vocab}}(n_t + 1),$$

where $n_t + 1$ slots are scored. (The extra `no-insert` class changes $|V_c|$ by $+1$, which is negligible at $|V_c| = 50\text{k}$.) AR baseline: $F_{\text{infer}}^{\text{AR}}(128) \approx 15.94\,\text{G}$. Take $n_t \in \{1, 2, 4, 8, 16, 32, 64, 128\}$. Then:

$$\sum_t F_{\text{transformer}}^{\text{full}}(n_t) \approx 22.06\,\text{G}, \qquad \sum_t F_{\text{vocab}}(n_t + 1) \approx 10.10\,\text{G},$$

so

$$F_{\text{infer,bal}}^{\text{InsT}}(128) \approx 32.16\,\text{G}, \qquad I_{\text{infer,bal}}^{\text{InsT}} = 2.02.$$

Take $n_t \in \{1, 2, \ldots, 128\}$. Then:

$$\sum_t F_{\text{transformer}}^{\text{full}}(n_t) \approx 714.26\,\text{G}, \qquad \sum_t F_{\text{vocab}}(n_t + 1) \approx 321.95\,\text{G},$$

so

$$F_{\text{infer,ser}}^{\text{InsT}}(128) \approx 1{,}036.20\,\text{G}, \qquad I_{\text{infer,ser}}^{\text{InsT}} = 65.01.$$

C.4.3 One-shot NAT (fertility / length latents): Non-Autoregressive NMT (Gu et al., 2018)

Gu et al. (2018) propose one-shot non-autoregressive translation (NAT), which predicts all target tokens in parallel by using discrete fertility/length latents and a single encoder–decoder pass.

For this numerical instantiation we use $n_{\text{src}} = n = 128$. Let K_{fert} denote the number of discrete fertility classes (the maximum fertility bucketed by the model), so each source token predicts a fertility $f_i \in \{0, 1, \ldots, K_{\text{fert}} - 1\}$; here we set $K_{\text{fert}} = 50$.

One-shot NAT runs a single encoder–decoder evaluation:

$$F_{\text{infer}}^{\text{NAT}}(n, n_{\text{src}}) \triangleq F_{\text{enc}}(n_{\text{src}}) + F_{\text{mult}}(n_{\text{src}}, d, K_{\text{fert}}) + \Big(F_{\text{transformer}}^{\text{full}}(n) + F_{\text{xattn}}(n, n_{\text{src}})\Big) + F_{\text{vocab}}(n).$$

With $n = n_{\text{src}} = 128$:

$$F_{\text{xattn}}(128, 128) \approx 3.93\,\text{G}, \qquad F_{\text{fert}}(128) = F_{\text{mult}}(128, 768, 50) \approx 4.92\,\text{M}.$$

Therefore:

$$F_{\text{infer}}^{\text{NAT}}(128, 128) \approx 31.19\,\text{G}, \qquad I_{\text{infer}}^{\text{NAT}} = 1.96.$$

C.4.4 Iterative Mask Refinement: Mask-Predict / CMLM (Ghazvininejad et al., 2019)

Ghazvininejad et al. (2019) start Mask-Predict from a fully-masked target and perform T_{mp} refinement cycles, each time masking low-confidence tokens and re-predicting them in parallel.

We instantiate $T_{\text{mp}} = 10$ and the common linear decay schedule:

$$m_0 = n, \qquad m_t = \left\lfloor n \cdot \frac{T_{\text{mp}} - t}{T_{\text{mp}}} \right\rfloor \quad (t = 1, \ldots, T_{\text{mp}} - 1).$$

For $n = 128$, $T_{\text{mp}} = 10$:

$$(m_0, \ldots, m_9) = (128, 115, 102, 89, 76, 64, 51, 38, 25, 12), \qquad \sum_{t=0}^{9} m_t = 700.$$

Method	Variant	$F_{\text{infer}}^{\mathcal{M}}$ (G)	$I_{\text{infer}}^{\mathcal{M}}$
AR	baseline	15.94	1
Reviser	$p_{\text{move}} \in \{0.20, 0.28, 0.33\}$	19.97–23.90	1.25–1.50
LevT	no EE, $R = 5$	154.62	9.70
LevT	no EE, $R = 10$	309.24	19.40
LevT	EE ($f_* = 0.5$), $R = 5$	110.17	6.91
LevT	EE ($f_* = 0.5$), $R = 10$	220.33	13.82
InsT	balanced-tree	32.16	2.02
InsT	serial	1,036.20	65.01
NAT	one-shot	31.19	1.96
CMLM	Mask-Predict, $T_{\text{mp}} = 10$	189.05	11.86

Table 10: Summary of inference compute under a matmul-dominant FLOPs proxy.

At each iteration we pay decoder trunk + cross-attention + vocab projection on only the masked tokens:

$$F_{\text{iter}}^{\text{CMLM}}(t, n, n_{\text{src}}) = F_{\text{transformer}}^{\text{full}}(n) + F_{\text{xattn}}(n, n_{\text{src}}) + F_{\text{vocab}}(m_t).$$

Encoder cost is paid once: $F_{\text{enc}}(n_{\text{src}}) = F_{\text{transformer}}^{\text{full}}(n_{\text{src}})$.

With $n_{\text{src}} = n = 128$:

$$F_{\text{transformer}}^{\text{full}}(128) + F_{\text{xattn}}(128, 128) \approx 15.10\,\text{G}.$$

Masked-only vocab projection across all iterations:

$$\sum_{t=0}^{T_{\text{mp}}-1} F_{\text{vocab}}(m_t) \approx 26.88\,\text{G}.$$

Therefore:

$$F_{\text{infer}}^{\text{CMLM}}(128, 128) = F_{\text{enc}}(128) + \sum_{t=0}^{T_{\text{mp}}-1} F_{\text{iter}}^{\text{CMLM}}(t, 128, 128) \approx 189.05\,\text{G},$$

$$I_{\text{infer}}^{\text{CMLM}} = 11.86.$$

C.4.5 Summary table

D More Qualitative Examples

Table 11 gives a concrete obfuscation–restoration trajectory example used for qualitative inspection.

D.1 Obfuscation–Restoration Trajectory Example

The example referred to in this subsection is Table 11.

We illustrate how an obfuscation trajectory (left) can be inverted to form a restoration trajectory (right). The cursor is shown as a vertical bar "|" inside the bracketed canvas.

Concatenating the executed edit actions, the obfuscation trajectory is:

$$B = [\textsc{Delete}, \textsc{Move}(-1), \textsc{Delete}, \textsc{Insert}(\texttt{hello}), \textsc{Move}(+2), \textsc{Delete}, \textsc{Delete}, \textsc{Delete}, \textsc{Move}(+1), \textsc{Delete}].$$

Reversing the order and inverting each action yields the restoration trajectory:

$$\cancel{A} = [\textsc{Insert}(\texttt{lamb.}), \textsc{Move}(-1), \textsc{Insert}(\texttt{hello}), \textsc{Insert}(\texttt{had}), \textsc{Insert}(\texttt{little}), \textsc{Move}(-2), \textsc{Delete}, \textsc{Insert}(\texttt{Mary}), \textsc{Move}(+1), \textsc{Insert}(\texttt{a}), \textsc{Stop}].$$

Obfuscation (corrupt target → blank)	Restoration (invert & apply → target)
1. Initial [Mary had a| little lamb.]	1. Initial [|]
2. DELETE [Mary had| little lamb.]	2. INSERT(lamb.) [lamb.|]
3. MOVE(−1) [Mary| had little lamb.]	3. MOVE(−1) [|lamb.]
4. DELETE [|had little lamb.]	4. INSERT(hello) [hello| lamb.]
5. INSERT(hello) [hello| had little lamb.]	5. INSERT(had) [hello had| lamb.]
6. MOVE(+2) [hello had little| lamb.]	6. INSERT(little) [hello had little| lamb.]
7. DELETE [hello had| lamb.]	7. MOVE(−2) [hello| had little lamb.]
8. DELETE [hello| lamb.]	8. DELETE [|had little lamb.]
9. DELETE [|lamb.]	9. INSERT(Mary) [Mary| had little lamb.]
10. MOVE(+1) [lamb.|]	10. MOVE(+1) [Mary had| little lamb.]
11. DELETE [|]	11. INSERT(a) [Mary had a| little lamb.]
	12. STOP [Mary had a| little lamb.]

Table 11: Side-by-side obfuscation and restoration (richer-action-space variant for illustration). Restoration is obtained by reversing the obfuscation actions and inverting each step (Delete↔Insert of the deleted token, Move(Δ)↔Move(−Δ), Insert(token)↔Delete).

Applying $\not\leftarrow$ to the blank state deterministically reconstructs the original canvas shown on the left.

E Selected Ranked Responses and Restoration Trajectories

Interactive HTML visualizations will be released with the code repository after review; the static text and action trajectories below provide the review-version qualitative evidence.

E.1 100M Examples

100M Reviser Example 1

Text (prompt segment in blue).

Claudine started ballet at the age of three in Chapel Hill, North Carolina and continued at Pofahl Studios under the instruction of Kim Tuttle and Judy Skinner in Gainesville, Texas. She taught at various studios including New York City Dance Center, the Brooklyn School of Dance, and the Center for Arts & Dance at Syracuse Dance Studio in Little Rock, New York. Claudine began ballet at the age of three. She started teaching at the same time, and continued teaching at the age of two teaching. She also taught with several dance classes during her time in the private and private classes. She has taught classes both in the private and semi private areas, She has taught for years as an educator and has had to have more than as many dancers have taught.

Restoration trajectory (189 actions).

```
INSERT `Cl', INSERT `aud', INSERT `ine', INSERT ` started', INSERT ` ballet', INSERT ` at', INSERT ` the', INSERT ` age',
INSERT ` of', INSERT ` three', INSERT ` in', INSERT ` Chapel', INSERT ` Hill', INSERT `,', INSERT ` North', INSERT `
Carolina', INSERT ` and', INSERT ` continued', INSERT ` at', INSERT ` P', INSERT `of', INSERT `ahl', INSERT ` Studios',
INSERT ` under', INSERT ` the', INSERT ` instruction', INSERT ` of', INSERT ` Kim', INSERT ` Tut', INSERT `tle', INSERT `
and', INSERT ` Judy', INSERT ` Skinner', INSERT ` in', INSERT ` Gain', INSERT `.', [MOVE -1], INSERT ` have', INSERT `
taught', [MOVE -2], INSERT ` as', INSERT ` many', INSERT ` dancers', [MOVE -2], [MOVE -1], INSERT ` and', INSERT ` has',
INSERT ` had', INSERT ` to', [MOVE -4], INSERT `,', [MOVE -1], INSERT `,', INSERT `.', [MOVE -2], [MOVE +2], [MOVE -1],
```

[MOVE -1], INSERT ` New', INSERT ` York', INSERT `.', INSERT `\n', INSERT `Cl', INSERT `aud', INSERT `ine', INSERT ` began', [MOVE -8], INSERT ` in', INSERT ` Little', INSERT ` Rock', INSERT `,', [MOVE +8], INSERT ` ballet', INSERT ` at', INSERT ` She', INSERT ` started', INSERT ` teaching', INSERT ` at', [MOVE -4], INSERT ` the', INSERT ` age', INSERT ` of', INSERT ` three', INSERT `.', [MOVE +4], INSERT ` the', INSERT ` same', INSERT ` time', [MOVE +1], INSERT ` and', INSERT ` teaching', [MOVE +2], INSERT ` as', [MOVE +4], [MOVE -8], INSERT ` continued', [MOVE +4], INSERT ` an', INSERT ` educator', [MOVE -32], [MOVE +32], [MOVE -4], INSERT ` She', INSERT ` also', INSERT ` taught', INSERT ` with', INSERT ` several', INSERT ` dance', INSERT ` classes', INSERT ` during', INSERT ` her', INSERT ` time', INSERT ` in', INSERT ` the', INSERT ` private', INSERT ` private', INSERT ` classes', INSERT `.', INSERT ` She', INSERT ` has', INSERT ` taught', INSERT ` classes', INSERT ` both', INSERT ` in', INSERT ` the', INSERT ` private', INSERT ` and', INSERT ` semi', INSERT ` private', INSERT ` areas', [MOVE +8], INSERT ` more', INSERT ` than', [MOVE -2], INSERT ` have', [MOVE -8], [MOVE -16], INSERT ` and', [MOVE -16], INSERT ` teaching', INSERT ` at', INSERT ` the', INSERT ` age', INSERT ` of', INSERT ` two', [MOVE +32], INSERT ` She', INSERT ` has', INSERT ` taught', INSERT ` for', INSERT ` years', [MOVE -64], [MOVE -8], INSERT `esville', INSERT `,', INSERT ` Texas', INSERT `.', INSERT ` She', INSERT ` taught', INSERT ` at', INSERT ` various', INSERT ` Dance', INSERT ` Studio', [MOVE -2], INSERT ` studios', INSERT ` including', INSERT ` New', INSERT ` York', INSERT ` City', INSERT ` Dance', INSERT ` Center', INSERT `,', INSERT ` the', INSERT ` Brooklyn', INSERT ` School', INSERT ` of', INSERT ` Dance', INSERT `,', INSERT ` and', INSERT ` the', INSERT ` Center', INSERT ` for', INSERT ` Arts', INSERT ` &', INSERT ` Dance', INSERT ` at', INSERT ` Syracuse', [MOVE -32], STOP

100M Reviser Example 2

Text (prompt segment in blue).

Do you want to hire a trusted Professional Cleaners in Mitcham Wandsworth London SW17'
Hire our dependable Professional Cleaners company in Mitcham Wandsworth London Wandsworth for a free no-obligation quote. With 15 years of experience, we're a professional cleaning company and have experience with everything from carpeting, cleaning, cleaning to cleaning and so forth.
For a complete list of the services available, please contact us today.
If you'd like to hire, we'll be able to look at your work and bring you the best solution to your requirements. We will also assist you.
We know that working with an experienced man and team will make the key difference.

Restoration trajectory (186 actions).

INSERT `Do', INSERT ` you', INSERT ` want', INSERT ` to', INSERT ` hire', INSERT ` a', INSERT ` trusted', INSERT ` Professional', INSERT ` Clean', INSERT `ers', INSERT ` in', INSERT ` Mitch', INSERT `am', INSERT ` W', INSERT `ands', INSERT `worth', INSERT ` London', INSERT ` SW', INSERT `17', INSERT `'', INSERT `\n', INSERT `H', INSERT `ire', INSERT ` our', INSERT ` depend', INSERT `able', INSERT ` Professional', INSERT ` Clean', INSERT `ers', INSERT ` company', INSERT ` in', INSERT ` Mitch', INSERT `am', INSERT ` W', INSERT `ands', INSERT `.', [MOVE -1], [MOVE +1], [MOVE -1], [MOVE +1], [MOVE -1], INSERT ` the', [MOVE -1], INSERT `.', INSERT `\n', INSERT `We', INSERT ` know', [MOVE -2], [MOVE -2], INSERT ` you', [MOVE +4], [MOVE +1], INSERT ` key', INSERT ` difference', [MOVE -8], INSERT `'', INSERT `'', INSERT `ll', INSERT ` be', INSERT ` able', INSERT ` to', INSERT ` and', INSERT ` bring', INSERT ` you', INSERT ` your', INSERT ` requirements', INSERT `.', INSERT ` We', INSERT ` will', INSERT ` also', INSERT ` assist', [MOVE -16], [MOVE +8], [MOVE -2], INSERT ` look', INSERT ` at', INSERT ` work', [MOVE -1], INSERT ` your', [MOVE +16], INSERT ` that', INSERT ` working', INSERT ` with', INSERT ` an', INSERT ` experienced', INSERT ` man', INSERT ` and', INSERT ` team', INSERT ` will', INSERT ` make', [MOVE -8], [MOVE -16], [MOVE +4], [MOVE +2], [MOVE -4], INSERT ` the', INSERT ` best', INSERT ` solution', INSERT ` to', [MOVE -16], INSERT `d', INSERT ` like', INSERT ` to', INSERT ` hire', INSERT `,', INSERT ` we', INSERT `'', [MOVE -8], INSERT ` a', INSERT `\n', INSERT `If', INSERT ` you', INSERT `'', [MOVE -4], INSERT ` professional', INSERT ` services', [MOVE -1], INSERT ` the', [MOVE -2], [MOVE -1], INSERT `worth', INSERT ` London', INSERT ` W', INSERT `ands', INSERT `worth', INSERT ` for', INSERT ` a', INSERT ` free', INSERT ` no', INSERT `-', INSERT `ob', INSERT `lig', INSERT `ation', INSERT ` quote', INSERT `.', INSERT ` With', INSERT ` 15', INSERT ` years', INSERT ` of', INSERT ` experience', INSERT `,', INSERT ` we', INSERT `'', INSERT `'', INSERT `re', [MOVE +2], INSERT ` cleaning', INSERT ` company', INSERT ` with', INSERT ` everything', INSERT ` from', INSERT ` carpet', INSERT `ing', INSERT `,', INSERT ` cleaning', INSERT `,', INSERT ` cleaning', INSERT ` to', INSERT ` cleaning', INSERT ` and', INSERT ` so', INSERT ` forth', INSERT `.', INSERT `\n', INSERT `For', INSERT ` a', INSERT ` complete', INSERT ` list', INSERT ` of', [MOVE +2], INSERT ` available', INSERT `,', INSERT ` please', INSERT ` contact', INSERT ` us', INSERT ` today', INSERT `.', [MOVE +32], [MOVE -64], [MOVE +2], INSERT ` and', INSERT ` have', INSERT ` experience', [MOVE -1], [MOVE -32], STOP

100M Reviser Example 3

Text (prompt segment in blue).

5 beds | 3 baths | 2,340 sqft | $597 per sq. ft.
Run, dont walk. Come see this beautiful custom home near the beach!
This is a beautiful home, and this home has a beautiful fireplace in the main living room and a large open living room. There is a nice large dining room, a dining room, full kitchen, a large gas stove the master bathroom with and a living. The entire home features room a gorgeous outdoor dining room. The 2 bedrooms are spacious and open-plan, and open-plan! Great location,; open concept kitchen convenient in the community. Open floor plan, walk-in closet & den. The location! This is a small town, and is a short- walk from the beach. Come to enjoy!

Restoration trajectory (192 actions).

INSERT `5', INSERT ` beds', INSERT ` |', INSERT ` 3', INSERT ` baths', INSERT ` |', INSERT ` 2', INSERT `,', INSERT `340', INSERT ` sq', INSERT `ft', INSERT ` |', INSERT ` $', INSERT `597', INSERT ` per', INSERT ` sq', INSERT `.', INSERT ` ft', INSERT `.', INSERT `\n', INSERT `Run', INSERT `,', INSERT ` dont', INSERT ` walk', INSERT `.', INSERT ` Come', INSERT ` see', INSERT ` this', INSERT ` beautiful', INSERT ` custom', INSERT ` home', INSERT ` near', INSERT ` the', INSERT ` beach', INSERT `!', INSERT ` and', [MOVE -1], [MOVE +1], [MOVE -1], [MOVE +1], INSERT `-', INSERT `!', [MOVE -2], INSERT ` is', INSERT ` a', INSERT ` short', [MOVE -4], [MOVE +4], [MOVE -4], INSERT `,', [MOVE -1], INSERT ` is', INSERT ` a', INSERT ` small', INSERT ` town', [MOVE +2], [MOVE +4], INSERT ` to', INSERT ` enjoy', [MOVE -2], INSERT ` walk', INSERT ` from', INSERT ` the', INSERT ` beach', INSERT `.', INSERT ` Come', [MOVE -16], INSERT `!', INSERT ` This', [MOVE -2], INSERT `\n', INSERT `This', INSERT ` is', INSERT ` a', INSERT ` beautiful', INSERT ` home', INSERT `,', INSERT ` and', INSERT ` this', INSERT ` home', INSERT ` has', INSERT ` a', INSERT ` beautiful', INSERT ` fireplace', INSERT ` in', INSERT ` the', INSERT ` main', INSERT ` living', INSERT ` room', INSERT ` and', INSERT ` a', INSERT ` large', INSERT ` open', INSERT ` living', INSERT ` room', INSERT `.', INSERT ` There', INSERT ` is', INSERT ` a', INSERT ` nice', INSERT ` large', INSERT ` dining', INSERT ` room', INSERT `,', INSERT ` full', INSERT ` kitchen', INSERT `,', INSERT ` a', INSERT ` large', INSERT ` gas', INSERT ` stove', INSERT ` and', INSERT ` a', INSERT ` living', INSERT ` room', INSERT ` a', INSERT `

gorgeous', INSERT `!', INSERT ` Great', INSERT ` location', [MOVE -4], [MOVE +1], INSERT ` outdoor', INSERT ` dining', INSERT ` room', INSERT `.', INSERT ` The', INSERT ` 2', INSERT ` bedrooms', INSERT ` are', INSERT ` spacious', INSERT ` and', INSERT ` open', INSERT `-', INSERT `plan', INSERT `,', INSERT ` and', INSERT ` open', INSERT `-', INSERT `plan', [MOVE +2], INSERT ` location', INSERT `,', INSERT ` convenient', INSERT ` in', INSERT ` the', [MOVE -32], [MOVE +32], INSERT ` community', INSERT `.', INSERT ` Open', INSERT ` floor', INSERT ` plan', INSERT `,', INSERT ` walk', INSERT `-', INSERT `in', INSERT ` closet', INSERT ` &', INSERT ` den', INSERT `.', INSERT ` The', [MOVE -1], [MOVE +16], [MOVE -32], INSERT `;', INSERT ` open', INSERT ` concept', INSERT ` kitchen', [MOVE -32], INSERT ` the', INSERT ` master', INSERT ` bathroom', INSERT ` with', [MOVE +1], [MOVE +2], INSERT `.', INSERT ` The', INSERT ` entire', INSERT ` home', INSERT ` features', [MOVE -16], [MOVE -2], [MOVE -2], INSERT `,', INSERT ` a', INSERT ` dining', INSERT ` room', [MOVE -8], [MOVE +2], [MOVE -32], STOP

100M Reviser Example 4

Text (prompt segment in blue).

Being able to present the findings of research into how effective Cognitive Behavioural Therapy is in a digital and online setting - specifically for treating youth anxiety - and in comparison to a traditional mental therapy approach, it has led to new challenges for children with cognitive disorders: they are now at the forefront of psychological change; that the technology can make changes to behavior of their children, the social and behavioral problems. The authors report that brain-related research, or in the course of memory, can provide a comprehensive assessment of the effective therapy with their personal and psychological concerns, their own self, and this are a new challenge.
For more information on the full text of the authors' case studies, please contact first to share article(s) of their authors, and submitting a review by.

Restoration trajectory (228 actions).

INSERT `Being', INSERT ` able', INSERT ` to', INSERT ` present', INSERT ` the', INSERT ` findings', INSERT ` of', INSERT ` research', INSERT ` into', INSERT ` how', INSERT ` effective', INSERT ` Cognitive', INSERT ` Beh', INSERT `aviour', INSERT `al', INSERT ` Therapy', INSERT ` is', INSERT ` in', INSERT ` a', INSERT ` digital', INSERT ` and', INSERT ` online', INSERT ` setting', INSERT ` -', INSERT ` specifically', INSERT ` for', INSERT ` treating', INSERT ` youth', INSERT ` anxiety', INSERT ` -', INSERT ` and', INSERT ` in', INSERT ` comparison', INSERT ` to', INSERT ` a', INSERT `.', [MOVE -1], [MOVE +1], [MOVE -1], [MOVE +1], [MOVE -1], [MOVE +1], [MOVE -1], INSERT ` by', [MOVE -1], INSERT ` and', [MOVE +1], [MOVE -1], [MOVE -1], INSERT `\n', [MOVE -1], [MOVE +1], [MOVE -1], [MOVE +1], INSERT `For', [MOVE -2], [MOVE +2], [MOVE -2], INSERT `.', [MOVE -1], INSERT ` are', [MOVE -1], [MOVE +1], [MOVE -1], [MOVE +1], [MOVE -1], INSERT ` with', [MOVE -1], [MOVE +2], [MOVE -2], [MOVE +4], [MOVE +1], INSERT ` the', INSERT ` of', [MOVE -1], INSERT ` full', INSERT `,', INSERT ` please', [MOVE +2], INSERT ` a', INSERT ` review', [MOVE -8], INSERT ` more', INSERT ` information', INSERT ` on', [MOVE -4], [MOVE -4], INSERT ` therapy', [MOVE +2], [MOVE -1], [MOVE -2], INSERT ` can', [MOVE +2], [MOVE +1], INSERT ` a', INSERT ` new', INSERT ` challenge', [MOVE +8], INSERT ` text', INSERT ` of', INSERT ` the', INSERT ` authors', INSERT `'', INSERT `'', INSERT ` case', INSERT ` studies', [MOVE +4], INSERT ` submitting', [MOVE -2], INSERT ` their', INSERT ` authors', INSERT `,', [MOVE -4], INSERT ` contact', INSERT ` first', INSERT ` to', INSERT ` share', INSERT ` article', INSERT `(', INSERT `s', INSERT `)', [MOVE -32], INSERT ` provide', INSERT ` the', [MOVE +2], INSERT ` their', INSERT ` their', INSERT ` own', INSERT ` self', INSERT `,', INSERT ` and', INSERT ` this', [MOVE -4], [MOVE -8], [MOVE +2], INSERT ` a', INSERT ` comprehensive', INSERT ` assessment', INSERT ` of', [MOVE +1], INSERT ` effective', [MOVE +2], [MOVE +1], INSERT ` personal', INSERT ` and', INSERT ` psychological', INSERT ` concerns', INSERT `,', [MOVE -16], INSERT `.', INSERT ` The', INSERT ` authors', INSERT ` report', INSERT ` that', INSERT ` brain', INSERT `-', INSERT `related', INSERT ` research', INSERT `,', INSERT ` or', INSERT ` in', INSERT ` the', INSERT ` course', INSERT ` of', INSERT ` memory', INSERT `,', [MOVE -2], [MOVE -8], [MOVE +64], [MOVE -4], [MOVE -1], [MOVE -64], [MOVE -2], INSERT ` and', INSERT ` behavioral', INSERT ` problems', [MOVE -1], [MOVE -2], INSERT ` to', INSERT ` social', [MOVE -2], INSERT ` therapy', INSERT ` approach', INSERT `,', INSERT ` has', INSERT ` led', [MOVE +1], INSERT ` new', INSERT ` challenges', INSERT ` for', INSERT ` children', INSERT ` with', INSERT ` cognitive', INSERT ` disorders', INSERT `:', INSERT ` the', [MOVE +1], [MOVE -2], INSERT ` they', INSERT ` are', INSERT `,', [MOVE -1], INSERT ` now', INSERT ` at', INSERT ` the', INSERT ` forefront', INSERT ` of', INSERT ` psychological', INSERT ` change', INSERT `;', INSERT ` that', INSERT ` the', INSERT ` technology', INSERT ` can', INSERT ` make', INSERT ` changes', INSERT ` to', INSERT ` behavior', INSERT ` of', INSERT ` their', INSERT ` children', [MOVE -32], INSERT ` it', [MOVE -4], INSERT ` traditional', INSERT ` mental', [MOVE -2], STOP

100M Reviser Example 5

Text (prompt segment in blue).

This article describes the international safety standards that iPhone batteries meet. Underwriters Laboratory (UL) 2054: Covers safety of lithium-ion batteries in general use.
This is the standard of safety, but it is also strictly acceptable for batteries in general use. The Standard refers to some battery manufacturers which other manufacturers must rely on the standard as a result of the standard. On other hand, if an iPhone is used to use a battery to replace or replace an old battery, there is no better the more reliable way to use the standard. As with other standard and in of batteries, safety of battery and battery batteries use in the case of battery performance issues. The standards used in this standard are described as part of the strict quality of the standard.

Restoration trajectory (202 actions).

INSERT `This', INSERT ` article', INSERT ` describes', INSERT ` the', INSERT ` international', INSERT ` safety', INSERT ` standards', INSERT ` that', INSERT ` iPhone', INSERT ` batteries', INSERT ` meet', INSERT `.', INSERT `\n', INSERT `Under', INSERT `writers', INSERT ` Laboratory', INSERT ` (', INSERT `UL', INSERT `)', INSERT ` 20', INSERT `54', INSERT `:', INSERT ` Co', INSERT `vers', INSERT ` safety', INSERT ` of', INSERT ` lithium', INSERT `-', INSERT `ion', INSERT ` batteries', INSERT ` in', INSERT ` general', INSERT ` use', INSERT `.', INSERT `\n', INSERT `.', [MOVE -1], [MOVE +1], [MOVE -1], INSERT ` the', INSERT ` standard', [MOVE -2], [MOVE +1], [MOVE -1], INSERT ` the', [MOVE -1], [MOVE +1], [MOVE -1], INSERT `,', INSERT ` of', INSERT ` batteries', INSERT ` use', [MOVE -4], [MOVE +1], [MOVE -1], INSERT ` batteries', [MOVE +4], INSERT ` in', [MOVE +1], INSERT ` case', [MOVE -8], INSERT ` of', [MOVE +2], INSERT ` safety', [MOVE -4], INSERT `.', INSERT ` On', INSERT ` and', INSERT ` in', [MOVE +4], [MOVE -8], INSERT ` as', INSERT ` a', INSERT ` result', INSERT ` of', INSERT ` the', INSERT ` standard', [MOVE -4], [MOVE -2], INSERT ` other', INSERT ` manufacturers', INSERT ` must', INSERT ` rely', INSERT ` on', INSERT ` the', INSERT ` standard', [MOVE +8], INSERT ` the', [MOVE -16], INSERT `.', [MOVE -1], INSERT `,', INSERT ` but', INSERT ` it', INSERT ` is', INSERT ` also', INSERT ` strictly', INSERT ` acceptable', INSERT ` for', INSERT ` batteries', INSERT ` in', INSERT ` general', INSERT ` use', [MOVE +32], [MOVE -8],

INSERT ` battery', [MOVE -4], [MOVE +4], [MOVE -16], [MOVE -8], INSERT ` The', INSERT ` Standard', INSERT ` refers', INSERT ` to', INSERT ` some', INSERT ` battery', INSERT ` manufacturers', INSERT ` which', [MOVE -16], [MOVE -4], [MOVE -1], INSERT ` safety', [MOVE -1], INSERT ` the', INSERT ` standard', INSERT ` of', [MOVE -1], [MOVE -2], INSERT `This', INSERT ` is', [MOVE +32], [MOVE +8], INSERT ` other', INSERT ` hand', INSERT `,', INSERT ` if', INSERT ` an', INSERT ` iPhone', INSERT ` is', INSERT ` used', INSERT ` to', INSERT ` use', INSERT ` a', INSERT ` battery', INSERT ` to', INSERT ` replace', INSERT ` or', INSERT ` replace', INSERT ` an', INSERT ` old', INSERT ` battery', INSERT `,', INSERT ` there', INSERT ` is', INSERT ` no', INSERT ` better', [MOVE +1], INSERT ` more', INSERT ` reliable', INSERT ` way', INSERT ` to', INSERT ` use', INSERT ` the', INSERT ` standard', INSERT `.', INSERT `\n', INSERT `As', INSERT ` with', INSERT ` other', INSERT ` standard', [MOVE +8], INSERT ` and', INSERT ` battery', [MOVE +4], [MOVE +1], INSERT ` of', INSERT ` battery', INSERT ` performance', INSERT ` issues', INSERT `.', INSERT ` The', INSERT ` standards', INSERT ` used', INSERT ` in', INSERT ` this', INSERT ` standard', INSERT ` are', INSERT ` described', INSERT ` as', INSERT ` part', INSERT ` of', INSERT ` the', INSERT ` strict', INSERT ` quality', INSERT ` of', [MOVE -2], [MOVE -64], [MOVE -2], [MOVE +8], [MOVE +8], [MOVE -64], STOP

E.2 300M Examples

300M Reviser Example 1

Text (prompt segment in blue).

Home to 279 units, Scarborough Wood Condos offers one and two bedroom + den suites. They range in size from 860 sq ft to 1120 sq ft. Building amenities include a fitness center, a fitness center, and private parking available.
Information is deemed correct at the time of publishing and is subject to change. Real estate listings obtained from third party sources are for consumers' personal purchasing decisions and should not be relied upon for any purpose other than to identify prospective properties consumers may be interested in purchasing.
The MLS should correct floor plans as required by the listing agents; however, all information provided by the listing agent may be different. Not all properties are the same and the brokers may change. Information should be independently verified, accuracy and accuracy are provided. All information is supplied by the MLS®, a program of the MLS. We do not guarantee the accuracy of this information. Please call us today!
Copyright 2016. All Rights Reserved.

Restoration trajectory (230 actions).

INSERT `Home', INSERT ` to', INSERT ` 279', INSERT ` units', INSERT `,', INSERT ` Scarborough', INSERT ` Wood', INSERT ` Cond', INSERT `os', INSERT ` offers', INSERT ` one', INSERT ` and', INSERT ` two', INSERT ` bedroom', INSERT ` +', INSERT ` den', INSERT ` suites', INSERT `.', INSERT ` They', INSERT ` range', INSERT ` in', INSERT ` size', INSERT ` from', INSERT ` 8', INSERT `60', INSERT ` sq', INSERT ` ft', INSERT ` to', INSERT ` 1', INSERT `120', INSERT ` sq', INSERT ` ft', INSERT `.', INSERT ` Building', INSERT ` amenities', INSERT `.', [MOVE -1], [MOVE +1], [MOVE -1], [MOVE +1], [MOVE -1], INSERT `.', INSERT ` We', INSERT ` do', INSERT ` not', INSERT ` guarantee', INSERT ` the', INSERT ` accuracy', INSERT ` this', INSERT ` information', INSERT `.', INSERT ` Please', INSERT ` call', INSERT ` us', INSERT ` today', INSERT `!', INSERT `.', INSERT ` All', INSERT ` Rights', INSERT ` Reserved', [MOVE -4], [MOVE -8], INSERT ` of', [MOVE -8], INSERT ` the', INSERT ` MLS', [MOVE +16], INSERT `\n', INSERT `Copyright', INSERT ` 2016', [MOVE -4], [MOVE -16], [MOVE -1], [MOVE +1], [MOVE -1], INSERT ` is', INSERT ` supplied', INSERT ` by', [MOVE +2], INSERT `®,', INSERT ` a', INSERT ` MLS', [MOVE -8], INSERT ` information', [MOVE -1], INSERT ` provided', INSERT `.', INSERT `\n', INSERT `All', [MOVE -4], INSERT ` as', INSERT ` required', [MOVE -2], INSERT ` and', INSERT ` floor', [MOVE -2], [MOVE +2], INSERT ` plans', [MOVE +2], INSERT ` by', INSERT ` the', INSERT ` listing', INSERT ` agents', INSERT `;', INSERT ` however', INSERT `,', INSERT ` all', INSERT ` information', INSERT ` provided', INSERT ` by', INSERT ` the', INSERT ` listing', INSERT ` agent', INSERT ` and', INSERT ` the', INSERT ` brokers', INSERT ` may', INSERT ` change', INSERT `.', INSERT ` Information', INSERT ` should', INSERT ` be', INSERT ` independently', INSERT ` verified', INSERT `,', INSERT ` accuracy', INSERT ` and', INSERT ` accuracy', INSERT ` are', [MOVE -16], INSERT ` may', INSERT ` be', INSERT ` different', INSERT `.', INSERT ` Not', INSERT ` all', INSERT ` properties', INSERT ` are', INSERT ` the', INSERT ` same', [MOVE +32], [MOVE -4], INSERT ` program', INSERT ` of', INSERT ` the', [MOVE +4], [MOVE -64], INSERT ` available', INSERT `.', INSERT `\n', INSERT `Information', INSERT ` is', INSERT ` deemed', INSERT ` correct', [MOVE +1], INSERT ` is', INSERT ` subject', INSERT ` to', INSERT ` change', INSERT `.', INSERT ` Real', INSERT ` consumers', INSERT "'", INSERT ` personal', INSERT ` purchasing', INSERT ` decisions', INSERT ` and', INSERT ` should', INSERT ` not', INSERT ` be', INSERT ` relied', INSERT ` upon', INSERT ` for', INSERT ` any', INSERT ` purpose', INSERT ` other', INSERT ` than', INSERT ` to', INSERT ` identify', INSERT ` prospective', INSERT ` properties', INSERT ` consumers', INSERT ` may', INSERT ` be', INSERT ` interested', INSERT ` in', INSERT ` purchasing', INSERT `.', INSERT `\n', INSERT `The', INSERT ` MLS', INSERT ` should', INSERT ` correct', [MOVE -32], INSERT ` estate', INSERT ` listings', INSERT ` obtained', INSERT ` from', INSERT ` third', INSERT ` party', INSERT ` sources', INSERT ` are', INSERT ` for', [MOVE -16], INSERT ` at', INSERT ` the', INSERT ` time', INSERT ` of', INSERT ` publishing', [MOVE +4], [MOVE -16], INSERT ` include', INSERT ` a', INSERT ` parking', [MOVE -1], INSERT ` fitness', INSERT ` center', INSERT `,', INSERT ` a', INSERT ` fitness', INSERT ` center', INSERT `,', INSERT ` and', INSERT ` private', [MOVE -4], [MOVE +4], [MOVE -8], [MOVE -2], [MOVE -1], STOP

300M Reviser Example 2

Text (prompt segment in blue).

with the new year, new content comes along. The Argent Dawn will start its operations in the Plaguelands on the 23rd of January - unlocking new missions to complete.
We will be adding more games and activities in the coming months, so make sure that you follow our page on Facebook and Twitter in order to get more information on the mission. We will be happy to help you with the new content.
As well as these new missions, we look forward to continue working on this new mission for more people!
This is a great time for our customers. Thanks to everyone who has supported us.
As always, we are happy to use this opportunity to thank all those who have provided feedback, input and suggestions. Thank you.

Restoration trajectory (203 actions).

INSERT `with', INSERT ` the', INSERT ` new', INSERT ` year', INSERT `,', INSERT ` new', INSERT ` content', INSERT ` comes', INSERT ` along', INSERT `.', INSERT ` The', INSERT ` Argent', INSERT ` Dawn', INSERT ` will', INSERT ` start', INSERT ` its', INSERT ` operations', INSERT ` in', INSERT ` the', INSERT ` Pl', INSERT `ag', INSERT `uel', INSERT `ands', INSERT ` on', INSERT ` the', INSERT ` 23', INSERT `rd', INSERT ` of', INSERT ` January', INSERT ` -', INSERT ` unlocking', INSERT ` new', INSERT ` missions', INSERT ` to', INSERT ` complete', INSERT `.', [MOVE -1], [MOVE +1], [MOVE -1], [MOVE +1], [MOVE -1], [MOVE +1], [MOVE -1], [MOVE +1], [MOVE -1], [MOVE +1], [MOVE -1], [MOVE +1], [MOVE -1], [MOVE +1], [MOVE -1], [MOVE +1], [MOVE -1], INSERT ` you', [MOVE -1], [MOVE +2], [MOVE -2], INSERT ` Thank', [MOVE -1], INSERT ` to', INSERT ` use', INSERT `.', [MOVE -1], INSERT ` this', INSERT ` opportunity', INSERT ` to', INSERT ` thank', INSERT ` all', INSERT ` those', INSERT ` who', INSERT ` have', INSERT ` provided', INSERT ` feedback', INSERT `,', INSERT ` input', INSERT ` and', INSERT ` suggestions', [MOVE -16], INSERT ` our', [MOVE -1], INSERT ` for', [MOVE +1], INSERT ` customers', INSERT ` happy', [MOVE -4], INSERT ` time', [MOVE -1], INSERT ` this', INSERT ` new', INSERT ` mission', INSERT ` for', INSERT ` more', [MOVE +8], [MOVE -8], INSERT ` people', INSERT `!', INSERT `\n', INSERT ` is', INSERT ` a', INSERT ` great', [MOVE +4], INSERT `,', INSERT ` we', INSERT ` are', [MOVE +16], [MOVE -32], [MOVE -2], [MOVE +1], [MOVE -1], [MOVE +2], [MOVE -2], INSERT ` to', INSERT ` continue', [MOVE -2], INSERT ` with', INSERT ` the', INSERT ` new', INSERT ` content', INSERT `.', INSERT `\n', INSERT `As', INSERT ` well', INSERT ` as', INSERT ` these', INSERT ` new', INSERT ` missions', INSERT `,', INSERT ` we', INSERT ` look', INSERT ` forward', [MOVE -16], INSERT ` the', INSERT ` mission', INSERT `.', INSERT ` be', INSERT ` happy', INSERT ` to', INSERT ` help', INSERT ` you', [MOVE -8], INSERT ` on', [MOVE -1], INSERT ` more', INSERT ` information', [MOVE +4], INSERT ` We', INSERT ` will', [MOVE -8], INSERT `.', INSERT `\n', INSERT `We', INSERT ` will', INSERT ` be', INSERT ` adding', INSERT ` more', INSERT ` games', INSERT ` and', INSERT ` activities', INSERT ` in', INSERT ` the', INSERT ` coming', INSERT ` months', INSERT ` get', [MOVE +32], [MOVE -1], INSERT ` working', INSERT ` on', [MOVE +8], INSERT `This', [MOVE -1], [MOVE +8], INSERT `.', INSERT ` Thanks', INSERT ` to', INSERT ` everyone', INSERT ` who', INSERT ` has', INSERT ` supported', INSERT ` us', INSERT `.', INSERT `\n', INSERT `As', INSERT ` always', [MOVE -32], [MOVE -32], [MOVE +2], INSERT `,', INSERT ` so', INSERT ` make', INSERT ` sure', INSERT ` that', INSERT ` you', INSERT ` follow', INSERT ` our', INSERT ` page', INSERT ` on', INSERT ` Facebook', INSERT ` and', INSERT ` Twitter', INSERT ` in', INSERT ` order', INSERT ` to', [MOVE +2], [MOVE -32], STOP

300M Reviser Example 3

Text (prompt segment in blue).

Go from black screen from your security cameras to full coverage of your retail or commercial security system with our security camera repair service.
Our customers are at the center of our universe and our service is second to none. Security cameras come to us when their needs aren't met. We can also repair or replace entire systems or use additional security equipment. We are passionate about the safety and security of our customers, and we are proud of our reputation. If you need an solution to your security camera immediate repair problem, please contact us.
Our goal is to give you a professional, cost-effective and affordable solution that is suitable for you. Our professional technicians have the time and expertise to provide you with excellent solutions, and the best prices and solutions available. To learn more, visit our website to learn more about our company and your security solutions.

Restoration trajectory (200 actions).

INSERT `Go', INSERT ` from', INSERT ` black', INSERT ` screen', INSERT ` from', INSERT ` your', INSERT ` security', INSERT ` cameras', INSERT ` to', INSERT ` full', INSERT ` coverage', INSERT ` of', INSERT ` your', INSERT ` retail', INSERT ` or', INSERT ` commercial', INSERT ` security', INSERT ` system', INSERT ` with', INSERT ` our', INSERT ` security', INSERT ` camera', INSERT ` repair', INSERT ` service', INSERT `.', INSERT `\n', INSERT `Our', INSERT ` customers', INSERT ` are', INSERT ` at', INSERT ` the', INSERT ` center', INSERT ` of', INSERT ` our', INSERT ` universe', INSERT `.', [MOVE -1], [MOVE +1], [MOVE -1], INSERT ` you', [MOVE +1], [MOVE -2], INSERT `\n', [MOVE -1], INSERT ` and', [MOVE +1], INSERT `Our', INSERT ` goal', INSERT ` is', INSERT ` to', INSERT ` give', [MOVE +1], INSERT ` a', INSERT ` professional', INSERT `,', INSERT ` cost', INSERT ` solutions', [MOVE -8], [MOVE -4], INSERT `.', [MOVE -2], INSERT `,', [MOVE -1], INSERT ` our', INSERT ` customers', [MOVE +2], INSERT ` we', INSERT ` are', INSERT ` proud', INSERT ` of', INSERT ` our', INSERT ` reputation', INSERT `.', INSERT ` If', INSERT ` you', INSERT ` need', INSERT ` an', INSERT ` immediate', INSERT ` repair', INSERT `,', INSERT ` please', INSERT ` us', [MOVE -1], INSERT ` contact', [MOVE +1], [MOVE -4], INSERT ` problem', [MOVE -16], [MOVE -2], INSERT ` and', INSERT ` our', INSERT ` service', INSERT ` is', INSERT ` second', INSERT ` to', INSERT ` none', INSERT `.', INSERT ` Security', INSERT ` cameras', INSERT ` come', INSERT ` to', INSERT ` us', INSERT ` when', INSERT ` their', INSERT ` needs', INSERT ` aren', INSERT "'t", INSERT ` of', [MOVE -1], INSERT ` met', INSERT `.', INSERT ` We', INSERT ` can', INSERT ` also', INSERT ` repair', INSERT ` or', INSERT ` replace', INSERT ` entire', INSERT ` systems', INSERT ` or', INSERT ` equipment', INSERT `.', INSERT ` We', INSERT ` are', INSERT ` passionate', INSERT ` about', INSERT ` the', INSERT ` safety', INSERT ` and', INSERT ` security', [MOVE +32], [MOVE +4], [MOVE -1], INSERT `-', INSERT `effective', INSERT ` and', INSERT ` affordable', INSERT ` security', [MOVE -1], INSERT ` solution', INSERT ` that', INSERT ` is', INSERT ` suitable', INSERT ` for', INSERT ` you', INSERT `.', INSERT ` Our', INSERT ` professional', INSERT ` technicians', INSERT ` have', INSERT ` your', [MOVE -1], INSERT ` the', INSERT ` time', INSERT ` and', INSERT ` expertise', INSERT ` to', INSERT ` provide', INSERT ` you', INSERT ` with', INSERT ` excellent', INSERT ` solutions', INSERT `,', INSERT ` and', INSERT ` the', INSERT ` best', INSERT ` prices', INSERT ` and', INSERT ` solutions', INSERT ` available', INSERT `.', INSERT ` To', INSERT ` learn', INSERT ` more', INSERT `,', INSERT ` visit', INSERT ` our', INSERT ` our', INSERT ` company', INSERT ` and', [MOVE +1], [MOVE -4], INSERT ` website', INSERT ` to', INSERT ` learn', INSERT ` more', INSERT ` about', [MOVE -64], INSERT ` solution', INSERT ` to', INSERT ` your', INSERT ` security', INSERT ` camera', [MOVE +1], [MOVE -32], INSERT ` use', INSERT ` additional', INSERT ` security', [MOVE -32], STOP

300M Reviser Example 4

Text (prompt segment in blue).

Mobile optimization is one of the top priorities for any website now-a-days because of increased traffic & conversions from devices like mobiles & tablets. So, folks who wish to get their business on Google and Google+ can also boost the ranking of your site in the search engines which will increase the visibility & engagement of your website and result more traffic and revenue to your website. And, it will also get search engines to believe in their website.
Whether you are looking for a complete mobile optimization for your website or simply want to know more about our mobile optimisation services now, we will provide you with the best mobile internet and mobile optimization services in your area.
Our website optimization services are designed to ensure that your page is accessible in all

browsers and if your website is not working efficiently, we will work with you to improve your page speed.

Restoration trajectory (202 actions).

INSERT `Mobile', INSERT ` optimization', INSERT ` is', INSERT ` one', INSERT ` of', INSERT ` the', INSERT ` top', INSERT ` priorities', INSERT ` for', INSERT ` any', INSERT ` website', INSERT ` now', INSERT `-', INSERT `a', INSERT `-', INSERT `days', INSERT ` because', INSERT ` of', INSERT ` increased', INSERT ` traffic', INSERT ` &', INSERT ` conversions', INSERT ` from', INSERT ` devices', INSERT ` like', INSERT ` mob', INSERT `iles', INSERT ` &', INSERT ` tablets', INSERT `.', INSERT ` So', INSERT `,', INSERT ` folks', INSERT ` who', INSERT ` wish', INSERT ` optim', [MOVE -1], [MOVE +1], [MOVE -1], [MOVE +1], INSERT ` you', INSERT `.', [MOVE -1], [MOVE -2], INSERT ` our', INSERT ` mobile', [MOVE +1], [MOVE +2], [MOVE -2], INSERT `isation', INSERT ` services', INSERT ` now', INSERT `,', INSERT ` with', [MOVE -1], INSERT ` we', INSERT ` will', INSERT ` provide', INSERT ` you', INSERT ` with', INSERT ` the', INSERT ` best', INSERT ` mobile', INSERT ` internet', [MOVE -16], INSERT ` website', INSERT ` a', INSERT ` complete', INSERT ` website', INSERT ` or', INSERT ` simply', INSERT ` want', INSERT ` to', INSERT ` know', INSERT ` more', INSERT ` about', [MOVE +16], INSERT ` and', INSERT ` mobile', INSERT ` optimization', INSERT ` services', INSERT ` in', INSERT ` your', INSERT ` area', INSERT `.', INSERT `\n', INSERT `Our', INSERT ` website', INSERT ` optimization', INSERT ` services', INSERT ` are', INSERT ` designed', INSERT ` to', INSERT ` ensure', INSERT ` that', INSERT ` your', INSERT ` page', INSERT ` is', INSERT `,', INSERT ` we', INSERT ` will', INSERT ` work', [MOVE +2], INSERT ` to', INSERT ` improve', [MOVE -8], INSERT ` accessible', INSERT ` in', INSERT ` all', INSERT ` browsers', INSERT ` and', INSERT ` if', INSERT ` your', INSERT ` website', INSERT ` is', INSERT ` not', INSERT ` working', INSERT ` efficiently', [MOVE +8], INSERT ` your', INSERT ` page', INSERT ` speed', [MOVE -64], [MOVE -4], INSERT ` for', INSERT ` your', [MOVE -4], INSERT `.', INSERT `\n', INSERT `Whether', INSERT ` you', INSERT ` are', INSERT ` looking', INSERT ` for', [MOVE -8], INSERT ` to', INSERT ` get', INSERT ` their', INSERT ` business', INSERT ` on', INSERT ` Google', INSERT ` and', INSERT ` their', [MOVE -1], INSERT ` Google', INSERT `+', INSERT ` can', INSERT ` also', INSERT ` boost', INSERT ` the', INSERT ` ranking', INSERT ` of', INSERT ` in', [MOVE +1], [MOVE -2], INSERT ` your', INSERT ` site', INSERT ` in', INSERT ` the', INSERT ` search', INSERT ` engines', INSERT ` which', INSERT ` will', INSERT ` increase', INSERT ` the', INSERT ` visibility', INSERT ` &', INSERT ` engagement', INSERT ` of', INSERT ` your', INSERT ` website', INSERT ` and', INSERT ` result', INSERT ` more', INSERT ` traffic', INSERT ` and', INSERT ` revenue', INSERT ` to', INSERT ` your', INSERT ` website', INSERT `.', INSERT ` And', INSERT ` also', INSERT ` get', INSERT ` search', INSERT ` engines', INSERT ` to', INSERT ` believe', [MOVE -2], [MOVE -4], INSERT `,', INSERT ` it', INSERT ` will', [MOVE -16], [MOVE +32], [MOVE +1], [MOVE +1], INSERT ` mobile', INSERT ` optimization', [MOVE -64], [MOVE -1], STOP

300M Reviser Example 5

Text (prompt segment in blue).

Workplace Injury Lawyer | Robert P. Schuster, P.C.
A fundamental and appropriate expectation of any employer is that a safe workplace is provided for all employees in the workplace. This is essential, and we are available to assist you with your legal needs. We represent all types of clients in numerous criminal cases, including local, state, and federal.
Robert P. Schuster and his staff are business injury lawyers, personal injury legal, and other types of business injury and can help you in a number of different fields in every aspect of your legal career.
Contact Robert P. Schuster, P.C. at (888) 876-3088 right away.
© 2014 Robert C. Law Firm, Inc.

Restoration trajectory (171 actions).

INSERT `Work', INSERT `place', INSERT ` Injury', INSERT ` Law', INSERT `yer', INSERT ` |', INSERT ` Robert', INSERT ` P', INSERT `.', INSERT ` Sch', INSERT `uster', INSERT `,', INSERT ` P', INSERT `.', INSERT `C', INSERT `.', INSERT `\n', INSERT `A', INSERT ` fundamental', INSERT ` and', INSERT ` appropriate', INSERT ` expectation', INSERT ` of', INSERT ` any', INSERT ` employer', INSERT ` is', INSERT ` that', INSERT ` a', INSERT ` safe', INSERT ` workplace', INSERT ` is', INSERT ` provided', INSERT ` for', INSERT ` all', INSERT ` employees', INSERT `,', [MOVE -1], [MOVE +1], INSERT `.', [MOVE -1], [MOVE -1], INSERT `,', INSERT ` personal', INSERT ` injury', INSERT ` legal', [MOVE +1], INSERT ` Inc', [MOVE -2], INSERT `,', INSERT ` and', INSERT ` and', INSERT ` can', [MOVE -8], INSERT ` injury', INSERT ` lawyers', [MOVE -2], INSERT ` business', [MOVE +8], INSERT ` other', INSERT ` types', INSERT ` of', INSERT ` business', INSERT ` injury', [MOVE +2], INSERT ` help', INSERT ` you', INSERT ` in', INSERT ` a', INSERT ` number', INSERT ` of', INSERT ` different', INSERT ` fields', INSERT ` in', INSERT ` every', INSERT ` aspect', INSERT ` of', INSERT ` your', INSERT ` legal', INSERT ` career', INSERT `.', INSERT ` Robert', INSERT `uster', INSERT `,', INSERT ` P', INSERT `.', INSERT ` C', INSERT `.', INSERT ` Law', INSERT ` Firm', [MOVE -8], INSERT ` P', INSERT `.', INSERT ` Sch', [MOVE -4], INSERT `\n', INSERT `Contact', [MOVE +8], INSERT `C', INSERT `.', INSERT ` at', INSERT ` (', INSERT `888', INSERT `)', INSERT ` 8', INSERT `76', INSERT `-', INSERT `30', INSERT `88', INSERT ` right', INSERT ` away', INSERT `.', INSERT `\n', INSERT `©', INSERT ` 2014', INSERT ` Robert', [MOVE +4], [MOVE -64], INSERT ` are', INSERT ` local', INSERT `,', INSERT ` state', INSERT `,', INSERT ` and', INSERT ` federal', INSERT `.', INSERT `\n', INSERT `Robert', INSERT ` P', INSERT `.', INSERT ` Sch', INSERT `uster', INSERT ` and', INSERT ` his', INSERT ` staff', INSERT ` are', [MOVE -16], [MOVE -2], INSERT ` in', INSERT ` the', INSERT ` workplace', INSERT `.', INSERT ` This', INSERT ` is', INSERT ` essential', INSERT `,', INSERT ` and', INSERT ` we', [MOVE +1], INSERT ` available', INSERT ` to', INSERT ` assist', INSERT ` you', INSERT ` with', INSERT ` your', INSERT ` legal', INSERT ` needs', INSERT `.', INSERT ` We', INSERT ` represent', INSERT ` all', INSERT ` types', INSERT ` of', INSERT ` clients', INSERT ` in', INSERT ` numerous', INSERT ` criminal', INSERT ` cases', INSERT `,', INSERT ` including', [MOVE -32], STOP

F Reproducibility Checklist

This appendix records the key artifacts and settings needed to reproduce the reported quality, trajectory, and FLOPs results. The paper source is self-contained in this single main.tex file (no external appendix \input files are required).

F.1 Artifacts, data, and runtime

- Code repository: URL withheld for anonymous review (code, configs, and results will be released upon publication).
- Released checkpoints: URL withheld for anonymous review (checkpoints will be released upon publication).

- Environment: scripts/requirements.txt (Python package list for training, inference, evaluation, and HTML visualization scripts).
- Core runtime: PyTorch 2.11, CUDA 13, RTX 5090.
- Evaluation dataset: allenai/c4 (English validation split), GPT-2 tokenizer (use_fast=False), keep total token length in 144.00 to 216.00, fixed 35-token prompt, target total length 180.
- Evaluation set size: 1000 prompts per seed.
- Data release policy: code is released end-to-end, but raw training/evaluation data and internal checkpoints are not bundled; all scripts accept user-provided local file paths.
- Licensing and intended use: all datasets and models used are publicly available and used in accordance with their respective licenses and intended research use (C4 and FineWeb are distributed under ODC-BY; the baseline/evaluator/judge models are used under their released terms for research). Our released code and checkpoints are provided under the MIT license for research use, consistent with the access conditions of the artifacts used. C4 and FineWeb are derived from Common Crawl web text and may contain personal or offensive content; we perform no additional data collection or de-anonymization and rely on the dataset providers' filtering, and the Ethical Considerations section notes that model outputs can reflect biases present in this data.

F.2 End-to-end code path

- Training scripts: scripts/train/train_ar.py and scripts/train/train_reviser.py.
- Inference scripts: scripts/inference/run_ar_inference.py and scripts/inference/run_reviser_inference.py.
- Evaluation scripts: scripts/eval/evalppl.py, scripts/eval/arena_from_judgments.py, and scripts/eval/trajectory_stats.py.
- HTML restoration-trajectory viewer: scripts/viz/build_restoration_trajectory_html.py.
- Single-command orchestration: scripts/reproduce_paper.py.
- Usage examples and CLI commands are documented in scripts/README.md.

F.3 Model and training

- Reviser architecture: 100M uses $(L, d, h, d_{\text{ff}}) = (24, 512, 8, 2048)$; 300M uses $(26, 896, 14, 3584)$; RoPE and tied embeddings.
- Context limits: 100M $(T_{\max}, L_{\max}) = (255, 512)$; 300M $(511, 512)$.
- Action space: $\mathcal{M} = \{\pm1, \pm2, \pm4, \pm8, \pm16, \pm32, \pm64, \pm128, \pm256, \pm512\}$.
- Optimizer/training: AdamW ($\beta_1 = 0.9, \beta_2 = 0.95, \epsilon = 10^{-8}$, wd=0.1), cosine schedule with 2000-step warmup, bf16, dropout 0.1, grad clip 1.0.
- Learning rates: 100M peak/min $10^{-4}/2 \times 10^{-5}$; 300M peak/min $10^{-5}/10^{-6}$.
- Effective batching: 100M (bs 64, accum 1, seq 256); 300M (bs 16, accum 4, seq 512).
- Training budget: 30B processed tokens.
- Trajectory construction: obfuscation mix $p(\textsc{Delete}) = 0.8$, $p(\textsc{Move}) = 0.2$.

F.4 Hardware and FLOPs accounting

- Hardware: single RTX 5090 (32GB VRAM), single-node PCIe setup (no DDP/FSDP/pipeline parallelism).
- Software: PyTorch 2.11 + CUDA 13; PyTorch SDPA attention kernels; no external flash_attn.
- FLOPs protocol for Appendix metrics: dominant-matmul proxy in Appendix B with shared constants $n = 128$, $|V_c| = 50000$, $d = 768$, $L = 12$, $d_{\text{ff}} = 4d$, $K_{\max} = 16$.

F.5 Decoding and inference configuration

- Action selection: multinomial sampling with temperature $\tau = 0.9$, top-$k = 50$, and no top-p truncation.
- Stop criteria: stop on END_OF_RESPONSE (id 50260); decoding caps generation at 256 actions.
- Validity masking: enforce cursor bounds, canvas length bounds, and operator validity (e.g., DELETE invalid at cursor position 0; reserved tokens never trained/selected).

- Prompting/conditioning (if any): no special delimiter tokens; boundary is positional. Prompt prefix occupies positions $[0, 35)$ and continuation/editable region is $[35, 180)$. For SEDD/MDLM, prefix projection clamps $x[:, 0 : 35] = \text{prefix_ids}$ at every diffusion step; for Reviser, the prompt is provided as the first 35 inserted prompt tokens and continuation actions are generated thereafter.
- Diffusion baselines: SEDD and MDLM are decoded with 128 diffusion steps.

F.6 Evaluation protocol and reporting

- Primary quality metrics: evalPPL and with-input arena win rates; implementation/library: GPT-2 Large scoring + Skywork-Critic-8B arena judge.
- Aggregation: mean $\pm$ std over 3 seeds (123, 124, 125).