SOLVING ONE PROBLEM WITH ANOTHER: LINEAR-COST DIFFUSION FOR LONG-CONTEXT LANGUAGE MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-context language modeling needs a global view of the answer and economical token-level computation. We present LaLanD (Linear Attention Language Diffusion), which unifies discrete diffusion with linear attention: diffusion organizes global answer refinement across rounds, while a bidirectional RWKV-7 denoiser predicts every masked token from the full context. We connect the denoising objective to linear-cost sequence mixing and derive a cost–error criterion stating when another denoising round improves the generated distribution under fixed-partition stochastic sampling; Lean-verified proofs establish nested refinement and a training-coverage bound, and an enumerable task confirms the predicted regimes. Accordingly, we build a 3.3B model from RWKV-7 World3 with 30B tokens resampled from its pretraining corpus. It raises its initialization's 20-task LongBench mean from 11.8 to 19.5 with gains on 18 of 20 tasks, reaches a 68.3% ten-task average, and improves RACE, while retaining a fixed-size recurrent state and $O(K(H+N))$ mixing for context length H, answer length N, and K rounds.

1 INTRODUCTION

Long-context language modeling requires retaining distant evidence and using it to construct a coherent answer. Two families of models offer complementary approaches to this problem. Linear-attention and recurrent models process sequences through compact states with linear mixing cost (Katharopoulos et al., 2020; Yang et al., 2024; Peng et al., 2025), but their standard autoregressive decoders commit to one token at a time without revisiting emitted tokens. Discrete diffusion language models predict every answer token in parallel and refine the whole canvas over successive rounds (Austin et al., 2021; Sahoo et al., 2024; Nie et al., 2025), but full-sequence variants with dense softmax attention re-read the context at quadratic mixing cost in each round. Combining the two makes repeated full-context refinement affordable: diffusion updates the answer canvas across rounds, and a linear-cost sequence mixer supplies token predictions within each round.

We introduce LaLanD, *Linear Attention Language Diffusion* (Figure 1). The answer canvas holds visible tokens and masks. In each round a bidirectional RWKV-7 denoiser reads the context and the canvas in one linear-cost scan and predicts every masked position; the most confident predictions are committed and become evidence for the next round. Repeated full-context refinement therefore stays linear in sequence length, with a fixed-size recurrent state.

Theoretical unification of discrete diffusion and linear attention. We show that the clean-token denoising objective is conditional distribution estimation whose sequence mixer is free to choose, and that replacing softmax attention by a normalized linear kernel turns each denoising round into a linear-cost recurrent scan (Section 3.1). On this basis we view the model along two dynamical axes—recurrence propagates information across positions; diffusion updates the conditioning canvas across rounds—and derive a cost–error criterion that separates the dependence recovered by revealing new evidence from the error of the learned conditionals that use it (Section 3.2). Finite-law proofs checked in the Lean proof assistant establish that splitting reveal groups while preserving previously visible tokens cannot increase forward Kullback–Leibler (KL) divergence under true conditionals, and bound conditional error when training sufficiently covers the relevant reveal histories; an enumerable token

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

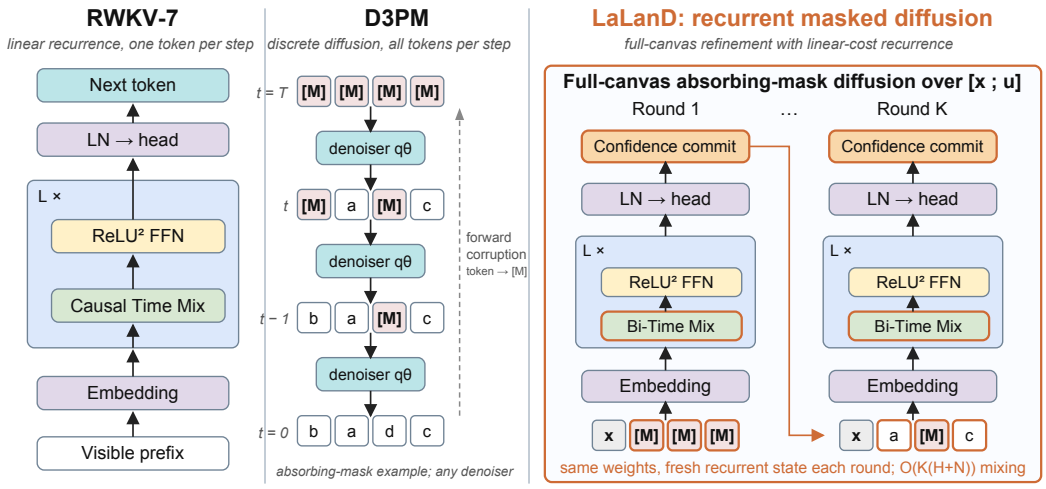


Figure 1: **From RWKV-7 and D3PM to LaLanD.** Left: RWKV-7 predicts one next token per step through a causal linear recurrence. Middle: an absorbing-mask instance of D3PM (Austin et al., 2021) corrupts tokens toward [M] (dashed arrow) and learns a reverse chain with a clean-token denoiser q_θ; the denoiser architecture is unconstrained. Right: LaLanD instantiates the D3PM denoiser as a bidirectional RWKV-7 stack over the full canvas $[\mathbf{x}; \mathbf{u}]$. Each round predicts all unresolved positions in one linear-cost scan and commits the most confident tokens, which condition the next round. $\mathbf{x}$ is the observed context, $\mathbf{u}$ the answer canvas, and [M] abbreviates [MASK]. The middle panel uses reverse diffusion time t, from a terminal index T to zero; LaLanD rounds are numbered forward from 1 to K. L is the block count, K the number of rounds, LN layer normalization, and FFN a feed-forward network. Orange marks what LaLanD changes relative to the RWKV-7 backbone; Figure 2 expands the blocks.

model exhibits the helpful, uninformative, and harmful regimes the criterion predicts (Appendix A, in particular A.2, A.3, and A.5).

LaLanD: efficient recurrent diffusion for long-context modeling. We translate this framework into a 3.3B model with full-canvas absorbing-mask diffusion, which replaces corrupted tokens with a mask, and a bidirectional RWKV-7 denoiser initialized from RWKV-7 World3 and adapted on 30B tokens resampled from its own pretraining corpus (Sections 3.3–3.5). On LongBench, LaLanD improves 18 of 20 tasks over its initialization and lifts the mean from 11.8 to 19.5; its ten-task capability average is 68.3%, and RACE also improves (Section 4). It does so while preserving the recurrent backbone's fixed-size state and $O(K(H+N))$ mixing cost for context length H, answer length N, and K denoising rounds, and it inherits the diffusion backbone's parallel decoding: every answer position is predicted in each round, so an N-token answer needs K sequential denoiser calls rather than N; each call rescans the full context, and on one H100 the result is 192 tokens per second against 14 for RWKV-7 (Sections 3.2 and 4.2).

2 Related Work

Discrete diffusion language models. D3PM formulates diffusion over categorical states using structured transition matrices, including an absorbing mask state (Austin et al., 2021). SEDD learns ratios of discrete data probabilities through score-entropy training (Lou et al., 2024). MDLM develops masked diffusion with a simplified denoising objective and a clean-token parameterization (Sahoo et al., 2024); LLaDA scales masked-token prediction to large language models (Nie et al., 2025). All of them use dense-attention denoisers.

Linear attention and state-space language models. Linear Transformers replace pairwise softmax attention with feature-map products whose sufficient statistics admit recurrent accumulation

(Katharopoulos et al., 2020). Gated linear attention makes memory retention input-dependent (Yang et al., 2024); RWKV develops recurrent token mixing, with RWKV-7 using generalized delta-rule state updates (Peng et al., 2023; 2025). In the state-space family, S4 models long-range sequences through structured dynamical systems (Gu et al., 2022). Mamba introduces input-dependent selective state updates, and Mamba-2 connects structured state-space computation with attention through state-space duality (Gu & Dao, 2024; Dao & Gu, 2024).

Combining linear sequence models with diffusion. DiffuMamba uses bidirectional Mamba denoisers for masked language modeling (Singh et al., 2026). Triplet-Block Diffusion RWKV combines RWKV recurrence with parallel bidirectional discrete diffusion (Lin et al., 2026). Linear-attention retrofits instead preserve softmax attention within an active denoising block while linearizing attention over preceding blocks (Kim et al., 2026).

Decoding schedules and error analysis. Beyond the choice of backbone, the reveal schedule determines which predictions become evidence for later rounds. Progressive Refinement Regulation learns a token-wise temperature controller from evolving denoising trajectories (Wan et al., 2026). Factorization-error analysis separates dependence lost through parallel sampling from errors in learned conditionals (Lavenant & Zanella, 2025).

3 METHOD

We first connect clean-token estimation to linear-cost sequence mixing (Section 3.1), then derive a cost–error criterion for additional refinement rounds (Section 3.2). Sections 3.3–3.5 instantiate this framework in a bidirectional RWKV-7 denoiser, its training objective, and the full-canvas generation loop (Figure 2).

3.1 CONDITIONAL TOKEN DIFFUSION

We use bold symbols for token sequences, vectors, and matrices; individual token values and scalar entries remain unbolded. Let H and N be the context and answer lengths, and let $\mathcal{V}$ be the token vocabulary. Write $\mathbf{x} = (x_1, \dots, x_H)$ for the observed context and $\mathbf{y} = (y_1, \dots, y_N) \in \mathcal{V}^N$ the target. An answer canvas $\mathbf{u} \in (\mathcal{V} \cup \{\texttt{[MASK]}\})^N$ contains either a visible token or a mask at each position. For a corruption mask $\mathbf{m} \in \{0,1\}^N$, define

$$u_i = \begin{cases} \texttt{[MASK]}, & m_i = 1, \\ y_i, & m_i = 0, \end{cases} \qquad q_\theta^i(v \mid \mathbf{x}, \mathbf{u}) = \mathrm{softmax}(f_\theta([\mathbf{x}; \mathbf{u}])_{H+i})_v. \tag{1}$$

Here f_θ is the denoiser with parameters θ; the semicolon denotes sequence concatenation, and its output contains vocabulary logits at each of the $S = H + N$ positions. The categorical probability $q_\theta^i(v \mid \mathbf{x}, \mathbf{u})$ predicts answer token v at position i, and $m_i = 1$ selects a prediction position. Independent masking with probability ρ is the elementary absorbing corruption $q_\rho(u_i \mid y_i) = (1-\rho)\mathbf{1}[u_i = y_i] + \rho\mathbf{1}[u_i = \texttt{[MASK]}]$ (Austin et al., 2021; Sahoo et al., 2024). This is a D3PM-style absorbing-mask diffusion process with direct clean-token prediction; training additionally samples span and prefix-completion masks (Section 3.4). Context tokens remain observed, only masked positions contribute to the denoising loss, and the denoiser receives no explicit diffusion-time embedding because the visible mask pattern provides corruption information. At inference the answer canvas starts entirely masked.

From the denoising objective to linear mixing. For a fixed corrupted canvas $\mathbf{z} = [\mathbf{x}; \mathbf{u}]$ and masked position i, let Y_i be the random clean token and $P_i(\cdot \mid \mathbf{z})$ its true conditional distribution. Writing H for entropy and KL for Kullback–Leibler divergence, the population cross-entropy satisfies

$$\mathbb{E}[-\log q_\theta^i(Y_i \mid \mathbf{z}) \mid \mathbf{z}] = \mathsf{H}(P_i(\cdot \mid \mathbf{z})) + \mathrm{KL}(P_i(\cdot \mid \mathbf{z}) \| q_\theta^i(\cdot \mid \mathbf{z})). \tag{2}$$

Thus the objective targets the clean-token conditional distribution; it leaves the sequence mixer open. A Transformer denoiser computes dense attention $\mathrm{softmax}(\mathbf{Q}\mathbf{K}^\top/\sqrt{d_k})\mathbf{V}$, where the rows of $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{S \times d_k}$ are query and key vectors, $\mathbf{V} \in \mathbb{R}^{S \times d_v}$ contains values, and d_k, d_v are the key and value dimensions, as in masked language diffusion (Sahoo et al., 2024; Nie et al., 2025). Choose a feature dimension r and map $\phi : \mathbb{R}^{d_k} \to \mathbb{R}^r$. Replace the exponential kernel by nonnegative weights

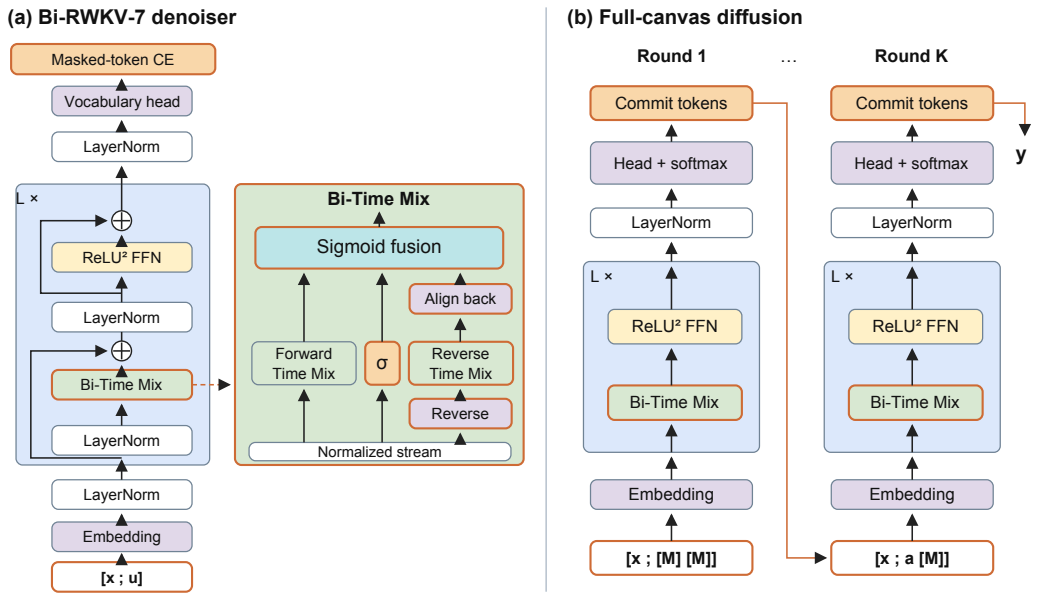


Figure 2: **LaLanD denoiser and diffusion loop.** (a) The bidirectional RWKV-7 denoiser, read bottom to top: the canvas $[\mathbf{x}; \mathbf{u}]$ enters the embedding and L residual blocks, then the vocabulary head and masked-token cross-entropy (CE). The auxiliary causal loss in Eq. (12) uses a separate forward-only pass on a clean sequence, which is not drawn. The expanded Bi-Time Mix runs forward and reversed RWKV-7 Time Mix branches on the same normalized stream, aligns the reverse output, and fuses the two by a sigmoid gate (Eqs. (8)–(9)). (b) Successive rounds call the same denoiser with fresh recurrent state; each round maps the current canvas to token probabilities and commits the most confident positions, and the updated canvas conditions the next round (Algorithm 1); the final round emits the completed answer $\mathbf{y}$. Panel (b) omits the residual connections and per-sublayer normalizations shown explicitly in panel (a). [M] abbreviates [MASK], LN denotes layer normalization, and FFN denotes a feed-forward network.

$\phi(\mathbf{q}_i)^\top \phi(\mathbf{k}_j)$, where $\mathbf{q}_i, \mathbf{k}_j$ are column vectors. With a positive row sum, the output vector $\mathbf{o}_i \in \mathbb{R}^{d_v}$ satisfies

$$\mathbf{o}_i^\top = \frac{\phi(\mathbf{q}_i)^\top \mathbf{C}}{\phi(\mathbf{q}_i)^\top \mathbf{c}}, \qquad \mathbf{C} = \sum_j \phi(\mathbf{k}_j)\mathbf{v}_j^\top, \quad \mathbf{c} = \sum_j \phi(\mathbf{k}_j). \tag{3}$$

The accumulator $\mathbf{C} \in \mathbb{R}^{r \times d_v}$ combines keys with value vectors $\mathbf{v}_j$, and $\mathbf{c} \in \mathbb{R}^r$ normalizes the kernel weights to sum to one. Associativity removes the pairwise attention matrix (Katharopoulos et al., 2020). In a causal scan, $\mathbf{C}$ and $\mathbf{c}$ are prefix accumulators; reverse scans provide the complementary direction. This identity is exact for the chosen feature kernel and motivates linear-cost denoising; the RWKV-7 mixer generalizes it with a learned, gated state recurrence.

3.2 Global Refinement with Linear-Cost Token Prediction

At fixed width and depth, each bidirectional recurrent mixing pass costs $O(H + N)$ in sequence length, so K denoising rounds cost $O(K(H + N))$ against $O(K(H + N)^2)$ for dense softmax mixing, and the recurrent state per direction and head has fixed size independent of $H + N$. Each round produces predictions for all N answer positions in one call, with no autoregressive dependency between their output tokens. The decoder uses K sequential full-canvas calls, fewer than the N autoregressive decoding steps when $K < N$; the default is $K = 16$. This counts sequential calls; Section 4.2 reports measured throughput. Total arithmetic per answer is $O(K(H + N))$ against $O(H + N)$ for a single causal pass, an additional factor of K in sequence-level scaling, with architecture-dependent constants; projections, the vocabulary head, and confidence selection add further work (Appendix A.4).

Sequence dynamics and denoising dynamics. For one direction and head, write the RWKV-7 state update as $\mathbf{M}_t = \mathbf{M}_{t-1}\mathbf{A}_t + \mathbf{J}_t$. Here $\mathbf{M}_t \in \mathbb{R}^{d_v \times d_k}$ is the recurrent state matrix, $\mathbf{A}_t \in \mathbb{R}^{d_k \times d_k}$ the input-dependent transition, and $\mathbf{J}_t \in \mathbb{R}^{d_v \times d_k}$ the write matrix at token position t, using per-head key and value dimensions. The perturbation $\Delta\mathbf{M}_t$ is the difference between two states at position t. Use the maximum absolute row-sum norm $\|\mathbf{M}\|_\infty = \max_a \sum_b |M_{ab}|$. Holding subsequent transitions and writes fixed, a state perturbation at position j propagates as

$$\Delta\mathbf{M}_t = \Delta\mathbf{M}_j \mathbf{A}_{j+1} \cdots \mathbf{A}_t, \qquad \|\Delta\mathbf{M}_t\|_\infty \le \|\Delta\mathbf{M}_j\|_\infty \prod_{r=j+1}^{t} \|\mathbf{A}_r\|_\infty. \tag{4}$$

The same products govern propagation of information from distant positions. Denoising adds a second evolution: newly revealed tokens change the input to the next full-sequence scan. Its error-control mechanism is therefore repeated conditional prediction on an updated canvas. The following decomposition quantifies when this reduces distributional error under the specified reveal process.

To isolate the effect of revealing additional information, consider a *fixed ordered partition* $\pi = (B_1, \ldots, B_K)$ of answer positions, chosen independently of sampled token values. Let $V_k = \bigcup_{j<k} B_j$ be the previously revealed positions and $\mathbf{y}_{V_k}$ their token values. For this analysis only, independently sample each position in B_k from its conditional marginal. With true answer law $P_{\mathbf{x}}$ on length-N token sequences and learned law

$$Q_{\theta,\pi}(\mathbf{y} \mid \mathbf{x}) = \prod_{k=1}^{K} \prod_{i \in B_k} q_\theta^i(y_i \mid \mathbf{x}, \mathbf{y}_{V_k}), \tag{5}$$

the KL chain rule gives

$$\mathrm{KL}(P_{\mathbf{x}} \| Q_{\theta,\pi}) = D_\pi(\mathbf{x}) + E_{\theta,\pi}(\mathbf{x}). \tag{6}$$

Here D_π is conditional dependence discarded by independent within-group sampling, and $E_{\theta,\pi}$ is the sum of errors in learned conditional marginals, both evaluated under target-distributed histories. This is established factorization-error accounting (Lavenant & Zanella, 2025); Appendix A.1 defines the terms and proves the identity. Write one for the single-group partition and multi for a given multi-group partition. Suppressing the common context $\mathbf{x}$, the same model has lower forward KL under the latter precisely when

$$E_{\theta,\mathrm{multi}} - E_{\theta,\mathrm{one}} < D_{\mathrm{one}} - D_{\mathrm{multi}}. \tag{7}$$

For an elementary group split, the recoverable dependence is a sum of conditional mutual informations (Appendix A.2). More generally, nested fixed schedules have nonincreasing forward KL when their predictions use the true conditionals (Proposition 3). For a learned denoiser, a reduction in discarded dependence helps only if conditional prediction error does not increase by more. Linear-cost recurrence makes additional passes potentially affordable; training must still support the longer contexts and reveal histories those passes encounter. Appendix A.3 further bounds conditional error in terms of population excess cross-entropy and training coverage: the training distribution must assign sufficient probability to the coordinate–history pairs queried by the target reveal process. This connects the reveal process to the denoiser's token-level excess risk; the coverage and risk conditions must hold for the relevant long contexts. Equation (7) applies to fixed-partition stochastic sampling. The confidence-ranked greedy implementation is evaluated separately in Section 4.

3.3 Bidirectional Recurrent Denoising

The network embeds $[\mathbf{x}; \mathbf{u}]$, applies L bidirectional RWKV-7 blocks of width d, and projects the final normalized features to vocabulary logits (Appendix C.1 gives the architecture dimensions). Let $\mathbf{H}_\ell \in \mathbb{R}^{d \times S}$ contain the input features of block ℓ, with positions as columns, and let $\mathbf{Z}_\ell = \mathrm{LN}_\ell(\mathbf{H}_\ell)$ be its layer-normalized features. The operator $\mathcal{R}$ reverses columns. With forward and reverse Time Mix modules $F_\ell^{\rightarrow}$ and $F_\ell^{\leftarrow}$, one block computes

$$\mathbf{O}_\ell^{\rightarrow} = F_\ell^{\rightarrow}(\mathbf{Z}_\ell), \qquad \mathbf{O}_\ell^{\leftarrow} = \mathcal{R}F_\ell^{\leftarrow}(\mathcal{R}\mathbf{Z}_\ell), \tag{8}$$

$$\mathbf{G}_\ell = \sigma(\mathbf{W}_\ell^g \mathbf{Z}_\ell + \mathbf{b}_\ell^g), \qquad \overline{\mathbf{H}}_\ell = \mathbf{H}_\ell + \mathbf{G}_\ell \odot \mathbf{O}_\ell^{\rightarrow} + (\mathbf{1} - \mathbf{G}_\ell) \odot \mathbf{O}_\ell^{\leftarrow}, \tag{9}$$

$$\mathbf{H}_{\ell+1} = \overline{\mathbf{H}}_\ell + \mathrm{FFN}_\ell(\mathrm{LN}'_\ell(\overline{\mathbf{H}}_\ell)). \tag{10}$$

The outputs $\mathbf{O}_\ell^{\rightarrow}$, $\mathbf{O}_\ell^{\leftarrow}$ and gate $\mathbf{G}_\ell$ have shape $d \times S$. The sigmoid σ acts elementwise; $\mathbf{W}_\ell^g \in \mathbb{R}^{d\times d}$ and $\mathbf{b}_\ell^g \in \mathbb{R}^d$ are learned gate parameters, with the bias broadcast across positions. The symbol $\odot$ denotes elementwise multiplication, $\mathbf{1}$ an all-ones array, and FFN a feed-forward network applied at each position. The first block additionally applies the pretrained input normalization. Each direction uses a complete RWKV-7 attention module with its own learned parameters and recurrent computation (Peng et al., 2025). Reverse-stream features are realigned before fusion; the FFN and normalizations operate once on the fused stream. Figure 2(a) expands this residual block into its forward and reverse Time Mix branches and their sigmoid fusion.

Initialization copies the pretrained causal attention weights into both directions and reuses the embedding, FFNs, normalizations, and vocabulary head. The two attention modules have separate parameter storage after cloning. With $\mathbf{W}_\ell^g = \mathbf{0}$ and $\mathbf{b}_\ell^g = 4 \cdot \mathbf{1}$, the initial forward weight is $\sigma(4) \approx 0.982$, so fusion starts close to the pretrained causal direction.

Every denoising round scans the entire current canvas anew with fresh recurrent state, so visible tokens on either side of an unresolved position inform its prediction in each round.

3.4 Training the token predictor

For a minibatch indexed by b, with token positions indexed by j, let the clean sequences be $\mathbf{s}^b$ and the corrupted sequences be $\tilde{\mathbf{s}}^b$. Let $\mathbf{m}^b$ be the binary supervision mask, with $m_j^b \in \{0, 1\}$ selecting eligible masked tokens. The objective normalizes by the number of masked tokens, rather than by sequence count:

$$\mathcal{L}_{\text{mask}} = -\frac{\sum_{b,j} m_j^b \log q_\theta^j(s_j^b \mid \tilde{\mathbf{s}}^b)}{\max(1, \sum_{b,j} m_j^b)}. \tag{11}$$

The distributed implementation aggregates the masked-token count across workers and scales local gradients accordingly. Padding and ineligible positions do not enter this loss. An auxiliary next-token loss on the forward branch keeps the pretrained causal ability:

$$\mathcal{L}_{\text{causal}} = -\frac{\sum_{b,j} e_{j+1}^b \log q_{\theta,\rightarrow}^j(s_{j+1}^b \mid \mathbf{s}_{\le j}^b)}{\max(1, \sum_{b,j} e_{j+1}^b)}, \qquad \mathcal{L} = \mathcal{L}_{\text{mask}} + \lambda_c \mathcal{L}_{\text{causal}}. \tag{12}$$

Here $q_{\theta,\rightarrow}^j$ is the forward-only network's next-token distribution at position j, computed on clean inputs; $e_{j+1}^b \in \{0, 1\}$ excludes padding and invalid targets, and λ_c weights the auxiliary loss.

Figure 2(a) shows the masked-token path. Corruption combines masking rates sampled per training window with contiguous-span masks; a prefix-completion component leaves a clean prefix and masks the remaining positions, exposing the denoiser to generation-shaped inputs. Adaptation enables gradients in stages: reverse attention and fusion gates first, upper-layer parameters next, then the whole network. Algorithm 2 summarizes the procedure and Appendix C.1 lists the settings.

3.5 Full-sequence generation

Figure 2(b) unfolds the inference loop: each vertical stack is one call to the same denoiser, and the connector between stacks carries committed tokens into the next answer canvas. Given $\mathbf{x}$ and an output budget N, set $\mathbf{u}^{(0)} = [\texttt{MASK}]^N$. At round $k \in \{1, \dots, K\}$, compute all-position predictions on $[\mathbf{x}; \mathbf{u}^{(k-1)}]$. For each unresolved answer position i, take

$$\hat{y}_i = \arg\max_{v \in \mathcal{V}_{\text{out}}} q_\theta^i(v \mid \mathbf{x}, \mathbf{u}^{(k-1)}), \qquad c_i = q_\theta^i(\hat{y}_i \mid \mathbf{x}, \mathbf{u}^{(k-1)}). \tag{13}$$

Here $\mathcal{V}_{\text{out}}$ is the output vocabulary excluding mask and padding IDs, $\hat{y}_i$ is the candidate token, and c_i its confidence. In Eq. (13), q_θ^i denotes the distribution renormalized over $\mathcal{V}_{\text{out}}$ after suppressing mask and padding IDs. If n_{k-1} positions have been committed, choose $b_k = \max(0, \lfloor Nk/K \rfloor - n_{k-1})$ highest-confidence unresolved positions. Write their predictions to the canvas and keep all previously visible tokens. The last round commits every remaining position. All predictions within a round use the same pre-commit canvas, so output tokens have no within-round autoregressive dependency; newly committed tokens condition the next round, and confidence selection can reveal tokens in any spatial order. The default uses $K = 16$, greedy selection, and no remasking; positive-temperature sampling may draw candidates before confidence ranking. Algorithm 1 gives the full procedure. Neither an observed context token nor a previously committed answer token is overwritten.

4 EXPERIMENTS

We evaluate both the practical benefits of recurrent diffusion and the mechanism predicted by the cost–error analysis. Table 5 in Appendix A.5 verifies the helpful, neutral, and harmful refinement regimes predicted by Eq. (7); Table 6 in Appendix A.6 shows how distance-balanced training reduces far-context conditional error in a controlled 455M study.

4.1 SETUP

LaLanD is a 3.3B-parameter model initialized from RWKV-7 World3 and adapted on 30B tokens resampled from that checkpoint's own pretraining corpus (Appendix C.1), so the paired comparison adds training exposure but no external corpus. The paired comparison evaluates the complete adaptation, including its objective, architecture, generation procedure, parameter increase, and training compute. Additional reference systems span autoregressive Transformers, recurrent/state-space models, and diffusion models with differing training exposure (Appendix C.2). Long-context evaluation uses LongBench (Bai et al., 2024) under the official script on the complete test sets of twenty tasks, together with RACE and bAbILong (Lai et al., 2017; Kuratov et al., 2024). The capability suite has thirteen task columns; Avg_{10} is the unweighted mean of MMLU, OpenBookQA, SocialIQA, Story Cloze, PIQA, ARC-e, SciQ, CommonsenseQA, LAMBADA, and RACE, SQuAD reports token F1, and the remaining columns report answer accuracy on 200 examples. Most bAbILong cells use 100 examples. Appendices C.3 and C.4 detail the capability and long-context protocols, respectively.

4.2 LONG-CONTEXT PERFORMANCE

LongBench. Table 1 compares LaLanD with its RWKV-7 initialization on twenty LongBench tasks. The 20-task mean rises from 11.79 to 19.52, a 66% relative gain; LaLanD is higher on 18 tasks, tied on one, and lower on one (VCSUM -3.3). Gains concentrate in single- and multi-document QA (2WikiMQA $+21.0$, MultiFieldQA-zh $+17.2$, NarrativeQA $+11.3$, MultiFieldQA-en $+10.4$, MuSiQue $+10.4$, HotpotQA $+9.3$), few-shot classification (TREC $+22.2$, LSHT $+8.3$), TriviaQA ($+13.7$), and code completion (RepoBench-P $+7.3$, LCC $+7.1$). Among summarization tasks, GovReport and QMSum improve, MultiNews is nearly flat, and VCSUM drops; PassageCount rises from 5.56 to 10.43, and PassageRetrieval-zh stays at zero for both systems.

Table 1: **LongBench under the official protocol.** Full test sets, official prompts, and official per-task metrics (F1 for the single- and multi-document QA tasks except DuReader, and for TriviaQA; ROUGE-L for DuReader, SAMSum, and the summarization tasks; accuracy for TREC, LSHT, and the passage tasks; edit similarity for code) for the RWKV-7 World3 initialization and LaLanD. Δ is LaLanD minus RWKV-7 in metric points and $\Delta\%$ the relative change; bold marks the higher score.

Task	RWKV-7	LaLanD	Δ	$\Delta\%$	Task	RWKV-7	LaLanD	Δ	$\Delta\%$
NarrativeQA	7.33	**18.67**	+11.34	+154.7%	TREC	11.11	**33.33**	+22.22	+200.0%
Qasper	7.11	**7.34**	+0.23	+3.2%	TriviaQA	78.57	**92.22**	+13.65	+17.4%
MultiFieldQA-en	9.98	**20.38**	+10.40	+104.2%	SAMSum	3.40	**10.41**	+7.01	+206.2%
MultiFieldQA-zh	3.71	**20.91**	+17.20	+463.6%	LSHT	8.33	**16.67**	+8.34	+100.1%
HotpotQA	6.06	**15.34**	+9.28	+153.1%	PassageCount	5.56	**10.43**	+4.87	+87.6%
2WikiMQA	7.86	**28.85**	+20.99	+267.0%	PassageRetrieval-zh	0.00	0.00	0.00	–
MuSiQue	0.00	**10.36**	+10.36	–	LCC	27.12	**34.25**	+7.13	+26.3%
DuReader	10.59	**12.59**	+2.00	+18.9%	RepoBench-P	20.44	**27.78**	+7.34	+35.9%
GovReport	8.09	**12.19**	+4.10	+50.7%					
QMSum	6.57	**7.85**	+1.28	+19.5%	*Aggregate over 20 tasks*				
MultiNews	7.01	**7.14**	+0.13	+1.9%	Mean	11.79	**19.52**	+7.73	+65.6%
VCSUM	**6.92**	3.64	−3.28	−47.4%	Wins / ties / losses				18 / 1 / 1

Reading and retrieval. On bAbILong QA1, LaLanD scores 44.0%, 19.0%, and 19.0% at 4k, 8k, and 16k, versus RWKV-7's 41.0%, 21.0%, and 17.0% (Table 2); QA2 at 4k improves from 19.0% to 20.0%, and QA1 at 8k drops from 21.0% to 19.0%. LaLanD's RACE score of 68.0% is the highest among the recurrent, state-space, and diffusion systems and third overall, behind Qwen2.5-3B and Qwen2.5-1.5B (Table 7).

Decoding efficiency. Table 3 measures the parallel decoding that LaLanD inherits from diffusion; each of the K rounds rescans the full context, so the speedup comes from $K \ll N$ sequential steps. On one H100 with a 512-token prompt and answer, LaLanD decodes at 192.3 tokens per second and finishes in 3.6 seconds including all rounds, against 13.6 tokens per second and 37.7 seconds for RWKV-7 and 12.7–18.6 tokens per second for the other baselines, at 8.4 GB peak memory (baselines 7.0–10.2 GB).

Table 2: **Reading and retrieval.** Answer accuracy (%) on RACE and bAbILong. Bold and underline mark the best and second-best among the systems shown; Table 8 adds Transformer references.

Task	GLA (1.4B)	Mamba-1 (2.8B)	RWKV-7 (2.9B)	LaLanD (3.3B)
RACE	19.0	24.0	<u>55.0</u>	**68.0**
bAbILong				
QA1 4k	14.0	2.0	<u>41.0</u>	**44.0**
QA1 8k	4.0	2.0	**21.0**	<u>19.0</u>
QA1 16k	2.0	0.0	<u>17.0</u>	**19.0**
QA2 4k	7.0	6.0	<u>19.0</u>	**20.0**

Table 3: **Decoding efficiency on one H100.** 512-token prompt and answer; time includes LaLanD's K rounds and memory is peak GPU allocation (Appendix C.5).

System	Mem. (GB)	tok/s	Time (s)
Qwen2.5 (3.1B)	7.7	15.5	33.1
Llama-3.2 (3.2B)	8.0	18.6	27.6
Phi-2 (2.7B)	7.4	18.0	28.5
Gemma-2 (2.6B)	9.9	15.6	32.9
Mamba-1 (2.8B)	10.2	12.7	40.3
RWKV-6 (3.0B)	**7.0**	15.9	32.2
RWKV-7 (2.9B)	7.1	13.6	37.7
LaLanD (3.3B)	8.4	**192.3**	**3.6**

4.3 Preserved short-context capability

Table 4 compares LaLanD with RWKV-6/7, GLA, Mamba, and the discrete-diffusion models MDLM and SEDD. The ten-task average is 68.3%, against 52.9% for the RWKV-7 World3 initialization, 36.2% for RWKV-6, and below 10% for the two diffusion baselines; the complete matrix with Transformer and continuous-representation references is in Table 7. So the long-context gains do not cost short-context accuracy ($+15.4$ points on Avg_{10}). ARC-c drops four points (Table 7).

Table 4: **Linear-attention, state-space, and discrete-diffusion comparison.** LaLanD uses a D3PM-style absorbing-mask diffusion process with a bidirectional RWKV-7 denoiser. Task cells are answer accuracy (%); Avg_{10} is the ten-task mean defined in Section 4.1, which adds RACE (reported in Table 7) to the nine task columns shown. Bold and underline mark the best and second-best scores among the rows shown; the complete reference matrix is in Table 7.

Model	Total	MMLU	OBQA	SIQA	StoryC.	PIQA	ARC-e	SciQ	CQA	LAMB.	Avg_{10}
Linear-attention and state-space language models											
GLA-1.3B	1.4B	10.5	13.0	24.0	26.0	44.0	31.0	25.5	17.0	45.5	25.6
RWKV-7 World3	2.9B	<u>36.5</u>	<u>43.0</u>	20.0	50.0	<u>61.0</u>	<u>75.5</u>	<u>72.0</u>	<u>39.5</u>	<u>76.0</u>	<u>52.9</u>
RWKV-6 World2.1	3.1B	27.0	32.0	<u>35.0</u>	<u>54.0</u>	48.0	26.0	22.0	20.0	73.0	36.2
Mamba-1	2.8B	27.0	16.0	20.0	20.5	7.5	27.5	33.5	21.5	43.0	24.1
Discrete diffusion language models											
MDLM-OWT	0.2B	3.5	3.5	3.0	7.0	2.5	0.5	0.5	2.5	20.5	4.5
SEDD-medium	0.4B	9.5	6.5	11.0	20.5	4.5	6.5	7.0	5.5	27.5	9.9
Bidirectional recurrent diffusion											
LaLanD	3.3B	**61.0**	**62.5**	**41.0**	**78.0**	**66.5**	**78.0**	**88.0**	**61.0**	**79.0**	**68.3**

5 Conclusion

LaLanD couples global diffusion refinement with linear-cost recurrent prediction for long-context language modeling. The cost–error analysis explains when additional reveals improve the generated distribution, with Lean-verified nested-refinement and training-coverage results and controlled validation of the underlying mechanism. The 3.3B realization improves 18 of 20 LongBench tasks over its

RWKV-7 initialization (mean 11.8 to 19.5), lifts the ten-task capability average from 52.9% to 68.3%, and decodes at 192 tokens per second on one H100, fourteen times the RWKV-7 rate, with a fixed-size recurrent state and $O(K(H+N))$ mixing. Repeated full-context refinement is therefore practical at 3.3B. The analysis covers fixed-partition stochastic sampling, and the comparison is against the initialization without a compute-matched causal control; both are the obvious next experiments.

ETHICS STATEMENT

This work adapts and evaluates language models on public corpora and public benchmarks; it involves no human subjects, private data, or new dataset release. The models studied here inherit the biases and misuse potential of their pretraining corpora and base checkpoints, and this work adds no mitigation beyond those of the underlying systems.

REPRODUCIBILITY STATEMENT

Appendix A states the assumptions and proofs of every theoretical claim; the Lean development and the finite-law enumeration of Appendix A.5 are included with the submitted source. Appendix B gives the generation and training algorithms; Appendix C specifies the architecture, initialization, adaptation recipe, and evaluation protocols; and Appendix D reports the complete benchmark tables.

AI USE STATEMENT

We used generative AI for source-code inspection, manuscript and LaTeX editing, and vector-figure preparation. The authors reviewed the text and figure sources, checked benchmark tables against recorded outputs, executed the included finite-law scripts, and checked the formal proofs with the Lean kernel. We take responsibility for all final content, including AI-assisted text, claims, and artifacts.

REFERENCES

Allen Institute for AI. OLMo-2-0425-1b model card. Official model card, 2025. URL https://huggingface.co/allenai/OLMo-2-0425-1B.

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

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137, 2024. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the International Conference on Machine Learning*, volume 202, pp. 2397–2430, 2023. URL https://proceedings.mlr.press/v202/biderman23a.html.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020. doi: 10.1609/aaai.v34i05.6239. URL https://ojs.aaai.org/index.php/AAAI/article/view/6239.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Proceedings of the International Conference on Machine Learning*, volume 235, pp. 10041–10071, 2024. URL https://proceedings.mlr.press/v235/dao24a.html.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, et al. Gemma 2: Improving open language models at a practical size. Technical report, Google DeepMind, 2024. URL https://storage.googleapis.com/deepmind-media/gemma/gemma-2-report.pdf.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=uYLFoz1vlAC.

Hongcan Guo, Qinyu Zhao, Yian Zhao, Shen Nie, Rui Zhu, Qiushan Guo, Feng Wang, Tao Yang, Hengshuang Zhao, Guoqiang Wei, and Yan Zeng. Continuous latent diffusion language model. *arXiv preprint arXiv:2605.06548*, 2026. URL https://arxiv.org/abs/2605.06548.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Keya Hu, Linlu Qiu, Yiyang Lu, Hanhong Zhao, Tianhong Li, Yoon Kim, Jacob Andreas, and Kaiming He. ELF: Embedded language flows. *arXiv preprint arXiv:2605.10938*, 2026. URL https://arxiv.org/abs/2605.10938.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proceedings of the International Conference on Machine Learning*, volume 119, pp. 5156–5165, 2020. URL https://proceedings.mlr.press/v119/katharopoulos20a.html.

Jinha Kim, Younghun Roh, and Jaeyeon Kim. Retrofitting linear attention into diffusion language models. In *COLM Workshop on Non-Autoregressive Language Models for Fast and Flexible Text Generation*, 2026. URL https://arxiv.org/abs/2608.06628. Accepted workshop paper; non-archival.

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. BABILong: Testing the limits of LLMs with long context reasoning-in-a-haystack. In *Advances in Neural Information Processing Systems*, volume 37, pp. 106519–106554, 2024. doi: 10.52202/079017-3381. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/c0d62e70dbc659cc9bd44cbcf1cb652f-Abstract-Datasets_and_Benchmarks_Track.html.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 785–794, 2017. doi: 10.18653/v1/D17-1082. URL https://aclanthology.org/D17-1082/.

Hugo Lavenant and Giacomo Zanella. Error bounds and optimal schedules for masked diffusions with factorized approximations. *arXiv preprint arXiv:2510.25544*, 2025. URL https://arxiv.org/abs/2510.25544.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, et al. DataComp-LM: In search of the next generation of training sets for language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 14200–14282, 2024. doi: 10.52202/079017-0455. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/19e4ea30dded58259665db375885e412-Abstract-Datasets_and_Benchmarks_Track.html.

Ke Lin, Yiyang Luo, Zhaolong Su, Yunya Song, and Anyi Rao. Triplet-block diffusion RWKV. In *Findings of the Association for Computational Linguistics: EMNLP 2026*, 2026. URL https://arxiv.org/abs/2605.25969.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the International Conference on Machine Learning*, volume 235, pp. 32819–32848, 2024. URL https://proceedings.mlr.press/v235/lou24a.html.

Justin Lovelace, Christian Belardi, Sofian Zalouk, Adhitya Polavaram, Srivatsa Kundurthy, and Kilian Q. Weinberger. Stop-think-autoregress: Language modeling with latent diffusion planning. In *Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=c05qIG1Z2B.

Meta. Llama 3.2 model card. Official model card, 2024. URL https://huggingface.co/meta-llama/Llama-3.2-3B.

Microsoft. Phi-2 model card. Official model card, 2023. URL https://huggingface.co/microsoft/phi-2.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, 2018. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 839–849, 2016. doi: 10.18653/v1/N16-1098. URL https://aclanthology.org/N16-1098/.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 50608–50646, 2025. doi: 10.52202/085713-1689. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/48b383b24230e0e6e649d9c98dae4d8c-Abstract-Conference.html.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1525–1534, 2016. doi: 10.18653/v1/P16-1144. URL https://aclanthology.org/P16-1144/.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, et al. RWKV: Reinventing RNNs for the transformer era. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14048–14077, 2023. doi: 10.18653/v1/2023.findings-emnlp.936. URL https://aclanthology.org/2023.findings-emnlp.936/.

Bo Peng, Daniel Goldstein, Quentin Anthony, Alon Albalak, Eric Alcaide, Stella Biderman, et al. Eagle and finch: RWKV with matrix-valued states and dynamic recurrence. In *Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=soz1SEiPeq.

Bo Peng, Ruichong Zhang, Daniel Goldstein, Eric Alcaide, Xingjian Du, Haowen Hou, Jiaju Lin, Jiaxing Liu, Janna Lu, William Merrill, Guangyu Song, Kaifeng Tan, Saiteja Utpala, Nathan Wilce, Johan S. Wind, Tianyi Wu, Daniel Wuttke, and Christian Zhou-Zheng. RWKV-7 "goose" with expressive dynamic state evolution. In *Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=ayB1PACN5j.

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, 2016. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264/.

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

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4463–4473, 2019. doi: 10.18653/v1/D19-1454. URL https://aclanthology.org/D19-1454/.

Vaibhav Singh, Oleksiy Ostapenko, Pierre-André Noël, Eugene Belilovsky, and Torsten Scholak. DiffuMamba: High-throughput diffusion LMs with Mamba backbone. In *ICLR Workshop on Multimodal Intelligence*, 2026. URL https://openreview.net/forum?id=4Q6TiBFiXy.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, 2019. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421/.

Evan Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, et al. 2 OLMo 2 furious. In *Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=2ezugTT9kU.

Lipeng Wan, Jianhui Gu, Junjie Ma, Jianguo Huang, Shiguang Sun, Siyuan Li, and Xuguang Lan. Progressive refinement regulation for accelerating diffusion language model decoding. *arXiv preprint arXiv:2603.04514*, 2026. URL https://arxiv.org/abs/2603.04514v1.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 94–106, 2017. doi: 10.18653/v1/W17-4413. URL https://aclanthology.org/W17-4413/.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In *Proceedings of the International Conference on Machine Learning*, volume 235, pp. 56501–56523, 2024. URL https://proceedings.mlr.press/v235/yang24ab.html.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

A TOKEN DISTRIBUTIONS AND COST–ERROR ANALYSIS

A.1 FIXED-PARTITION GENERATION

Fix the observed context $\mathbf{x}$ and answer length N. Let $P_{\mathbf{x}}$ be the true law of the random answer $\mathbf{Y}$ on $\mathcal{V}_{\text{out}}^N$, and let $\pi = (B_1, \dots, B_K)$ be a partition of answer coordinates, fixed independently of the values sampled during generation. The visible set before group k is $V_k = \bigcup_{j<k} B_j$. A denoiser call receives $\mathbf{x}$, the values $\mathbf{y}_{V_k}$, and masks at all other answer positions. Equation (5) defines a normalized distribution: summing the last group's independent categorical factors yields one, and repeating backwards eliminates every group. The analysis concerns stochastic token sampling under a fixed partition; a deterministic argmax decoder yields a point-mass law for which forward KL is not finite in general.

Define the discarded dependence and estimation terms by

$$D_\pi(\mathbf{x}) = \sum_k \mathbb{E}_{\mathbf{Y}_{V_k} \sim P_{\mathbf{x}}} \mathrm{KL}\left(P_{\mathbf{x}}(\mathbf{Y}_{B_k} \mid \mathbf{Y}_{V_k}) \,\middle\|\, \prod_{i \in B_k} P_{\mathbf{x}}(Y_i \mid \mathbf{Y}_{V_k}) \right), \tag{14}$$

$$E_{\theta,\pi}(\mathbf{x}) = \sum_k \sum_{i \in B_k} \mathbb{E}_{\mathbf{Y}_{V_k} \sim P_{\mathbf{x}}} \mathrm{KL}\big(P_{\mathbf{x}}(Y_i \mid \mathbf{Y}_{V_k}) \,\big\|\, q_\theta^i(Y_i \mid \mathbf{x}, \mathbf{Y}_{V_k}) \big). \tag{15}$$

The context is implicit in all $P_{\mathbf{x}}$ conditionals. Both expectations use target histories, not sampled model histories.

Proposition 1 (Factorized reveal error)**.** *For a finite answer alphabet, let $P_{\mathbf{x}}$ be any normalized nonnegative target law, and assume all learned categorical conditionals are strictly positive and normalized. Then* $\mathrm{KL}(P_{\mathbf{x}} \| Q_{\theta,\pi}) = D_\pi(\mathbf{x}) + E_{\theta,\pi}(\mathbf{x})$.

Proof. At any zero-probability history, choose a normalized conditional law; its contribution to a target-weighted expectation is zero. Factorize $P_{\mathbf{x}}(\mathbf{y})$ by the ordered groups and substitute Eq. (5) in $\mathbb{E}_{P_{\mathbf{x}}} \log(P_{\mathbf{x}}/Q_{\theta,\pi})$. For each group, insert and subtract $\sum_{i \in B_k} \log P_{\mathbf{x}}(y_i \mid \mathbf{y}_{V_k})$. Zero-mass outcomes contribute zero, with $0 \log 0 = 0$. On positive-mass outcomes, every relevant true marginal is positive. The group-joint-to-product ratio contributes Eq. (14). The remaining marginal ratios contribute Eq. (15). Summing gives the identity. □

Subtracting the one-call and multiple-call identities proves Eq. (7). Singleton reveal groups have $D_\pi = 0$; total error then equals conditional prediction error. Changing the reveal groups also changes the histories on which the model is queried.

A.2 INFORMATION RECOVERED BY AN ELEMENTARY REVEAL SPLIT

Fix an already visible history ξ (its visible coordinates and token values) and two disjoint groups of unresolved coordinates A and C. Let P_ξ be their normalized joint target law. Consider two samplers that use the true coordinate conditionals: Q_c reveals both groups simultaneously, while Q_s reveals A before C:

$$Q_c(\mathbf{y}_A, \mathbf{y}_C \mid \xi) = \prod_{i \in A \cup C} P_\xi(y_i), \tag{16}$$

$$Q_s(\mathbf{y}_A, \mathbf{y}_C \mid \xi) = \prod_{i \in A} P_\xi(y_i) \prod_{i \in C} P_\xi(y_i \mid \mathbf{y}_A). \tag{17}$$

Both laws are normalized. Define the information that the revealed group provides about coordinate i by

$$I_\xi(Y_i; \mathbf{Y}_A) = \mathbb{E}_{\mathbf{Y}_A \sim P_\xi} \mathrm{KL}(P_\xi(Y_i \mid \mathbf{Y}_A) \,\|\, P_\xi(Y_i)). \tag{18}$$

This is conditional mutual information at the fixed history ξ.

Proposition 2 (Gain from an elementary reveal split)**.** *The reduction in discarded dependence is*

$$\mathrm{KL}(P_\xi \| Q_c) - \mathrm{KL}(P_\xi \| Q_s) = \sum_{i \in C} I_\xi(Y_i; \mathbf{Y}_A) \geq 0. \tag{19}$$

Proof. On each positive-mass answer, subtract the two log likelihood ratios. The factors for group A cancel. Each remaining term is the log ratio between the true conditional of Y_i after revealing $\mathbf{Y}_A$ and its preceding marginal. Taking the target expectation gives Eq. (18), and Gibbs' inequality gives nonnegativity. Zero-mass answers contribute zero. The Lean development proves this identity for finite nonnegative targets. □

For a split inside a complete fixed schedule, every token outside C retains the same visible coordinates at its prediction round. Its true conditional factor therefore cancels from the complete likelihood ratio. Only the log ratios for C remain, weighted by the original target law. The full-sequence construction is normalized on original token assignments and has exactly this product of coordinate conditionals.

To state the general result, let $\mathcal{V}_i^\pi$ be the answer coordinates revealed before coordinate i under a fixed schedule π. Write $Q_\pi^\star$ for its true-conditional sampling law,

$$Q_\pi^\star(\mathbf{y} \mid \mathbf{x}) = \prod_{i=1}^{N} P_{\mathbf{x}}(y_i \mid \mathbf{y}_{\mathcal{V}_i^\pi}). \tag{20}$$

At zero-mass histories, use a uniform conditional distribution; these histories contribute zero to target expectations.

Proposition 3 (Nested fixed-schedule refinement)**.** *If $\mathcal{V}_i^\pi \subseteq \mathcal{V}_i^{\pi'}$ for every coordinate i, then*

$$\mathrm{KL}(P_{\mathbf{x}} \| Q_{\pi'}^\star) \le \mathrm{KL}(P_{\mathbf{x}} \| Q_\pi^\star). \tag{21}$$

Consequently, any finite chain of ordered group splits has nonincreasing forward KL under true-conditional sampling.

Proof. Both laws in Eq. (20) are normalized by successively summing the last group's categorical predictions. Subtract the full-sequence log likelihood ratios and sum over coordinates. For each i, the coarser visible assignment is a restriction of the finer one. Regrouping the target expectation by the finer assignment turns that coordinate's log ratio into the KL from its finer conditional to its coarser conditional, weighted by the target evidence probability. Each term is nonnegative by Gibbs' inequality. A split preserves all previous evidence, so this inclusion holds at every step; applying the inequality successively proves the chain statement. The Lean development proves the coordinate-law inequality and its repeated application. □

The right-hand side of Eq. (19) quantifies the useful dependence exposed by the split. With a learned denoiser, the change in conditional prediction error must also be included through Eq. (7); the finite-law example below varies this error independently of the reveal.

A.3 Training coverage and long-distance conditional error

For fixed $\mathbf{x}$ and N, let μ_π choose a coordinate uniformly from $\{1, \dots, N\}$ and draw its visible history under $P_{\mathbf{x}}$ and the fixed partition π. Let ν be a training-supervision law over the same coordinate–history space, with the same clean-token conditional law. For a visible history ξ and token coordinate i, write $\ell_\theta(\xi, i)$ for the KL from the true conditional at (ξ, i) to the denoiser, and $\mathcal{R}_\nu = \mathbb{E}_\nu \ell_\theta$ for population excess cross-entropy above conditional entropy.

The masked objective induces this supervision law by first sampling a realized minibatch, including its corruption, and then selecting one of its eligible masked positions uniformly. This reproduces the expected batch-normalized loss in Eq. (11). The resulting law of input–coordinate pairs induces ν after conditioning on $(\mathbf{x}, N)$. Batches without a masked position contribute zero; normalizing the supervision law conditions on a nonempty batch. Regrouping supervised events by input and coordinate decomposes their log loss into conditional entropy plus conditional KL. Under the shared clean-token conditional assumption above, the latter gives $\mathcal{R}_\nu$. The Lean development proves this decomposition.

Proposition 4 (Coverage-controlled conditional error)**.** *Suppose every coordinate–history pair with positive probability under μ_π also has positive probability under ν (written $\mu_\pi \ll \nu$), and its probability ratio $d\mu_\pi/d\nu$ is at most a finite constant κ. Then*

$$E_{\theta,\pi}(\mathbf{x}) \le N \kappa \mathcal{R}_\nu. \tag{22}$$

Proof. By uniform coordinate selection, $E_{\theta,\pi} = N\mathbb{E}_{\mu_\pi}\ell_\theta = N\mathbb{E}_\nu[(d\mu_\pi/d\nu)\ell_\theta]$. The density-ratio bound and nonnegativity of KL give the result. □

Together with Eq. (6), this yields $\mathrm{KL}(P_{\mathbf{x}}\|Q_{\theta,\pi}) \le D_\pi + N\kappa\mathcal{R}_\nu$. The bound relates population conditional risk to the target histories used by global refinement under the stated coverage assumption. Context length and evidence location enter through the conditional prediction task, separately from the output length N.

A.4 Cost accounting and a restricted quality–cost criterion

Let $C_M(\pi, \mathbf{x})$ be the request cost of model M under policy π, and suppose a budget B with $C_M(\pi, \mathbf{x}) \le B$ permits a recurrent model M_{rec} with a multiple-call policy and an attention model M_{att} with a one-call product policy, but no other policy for M_{att}. If

$$D_{\mathrm{multi}}(\mathbf{x}) + E_{M_{\mathrm{rec}},\mathrm{multi}}(\mathbf{x}) < D_{\mathrm{one}}(\mathbf{x}), \tag{23}$$

then M_{rec} has lower forward KL than that feasible one-call product sampler, because $E_{M_{\mathrm{att}},\mathrm{one}} \ge 0$.

For sequence length $S = H + N$, fixed depth L, width d, head geometry, and vocabulary size V, bidirectional recurrent mixing scales as $O(KS)$ in S. Token projections and FFNs contribute $O(KLSd^2)$ arithmetic, and a dense vocabulary head contributes $O(KSdV)$. To separate these charges from scheduling, let c_{arch} denote the per-position arithmetic charge at fixed architecture dimensions, including the head and vocabulary-wise processing. Let R_j denote confidence-selection work in round j. With K rounds, the corresponding arithmetic charge $T(K, S)$ is

$$T(K, S) = KS\, c_{\mathrm{arch}} + \sum_{j=1}^{K} R_j. \tag{24}$$

Confidence selection can use linear-time selection over N scores per round, adding $O(KN)$ work; a full sort instead adds $O(KN \log N)$. The Lean development formalizes these operation counts, which specify arithmetic scaling rather than device latency. Table 3 reports measured throughput and memory.

Relative to autoregressive decoding, which issues N sequential steps, LaLanD issues K full-canvas scans with $K = 16$ by default. Every scan predicts all N positions in one call, reducing the number of sequential calls when $K < N$; this does not remove the dependencies inside each recurrent scan or establish a latency speedup. Total arithmetic scales with K repeated passes. The per-direction, per-head recurrent state has fixed size independent of S; the token canvas, layer features, and the bSV vocabulary logits for batch size b scale linearly with S, as in any sequence model that materializes full-sequence outputs.

A.5 Analytical validation of the refinement criterion

We test Eq. (7) on a finite distribution whose dependence and conditional error can both be computed by enumeration. Let $\mathbf{Y} = (Y_0, Y_1, Y_2, Y_3) \in \{0, 1\}^4$. The anchor tokens Y_0 and Y_2 are independent fair bits. Conditional on the anchors, Y_1 copies Y_0 and Y_3 copies Y_2, independently, each with probability $1 - \eta$, where $\eta = 0.05$. Every one of the sixteen answers has positive probability. A predictor assigns probability $1/2$ when a token's partner is hidden and probability $1 - \epsilon$ to matching a visible partner. Here $\epsilon \in (0, 1)$ controls conditional prediction error.

One simultaneous reveal discards $D_{\mathrm{one}} = 2[\log 2 - \mathsf{H}_b(\eta)]$, where $\mathsf{H}_b(r) = -r \log r - (1-r)\log(1-r)$ is binary entropy in nats. Equation (19) identifies the dependence recovered by revealing $\{0, 2\}$ followed by $\{1, 3\}$: $D_{\mathrm{two}} = 0$. Its conditional error is

$$E_{\mathrm{two}} = 2\left[\eta \log \frac{\eta}{\epsilon} + (1 - \eta) \log \frac{1 - \eta}{1 - \epsilon}\right]. \tag{25}$$

Thus the two-call policy improves exactly when this error is below $2[\log 2 - \mathsf{H}_b(\eta)]$ (Table 5). By contrast, revealing $\{0, 1\}$ and then $\{2, 3\}$ gives no useful conditioning within either dependent pair; it retains the one-call dependence term despite using a second pass.

Reveal policy	ϵ	D	E	$\mathrm{KL}(P\|Q)$
One call	0.05	0.989264	0	0.989264
Useful two-call reveal	0.05	0	0	0
Useful two-call reveal	0.30	0	0.401049	0.401049
Useful two-call reveal	0.50	0	0.989264	0.989264
Useful two-call reveal	0.80	0	2.683216	2.683216
Uninformative two-call reveal	0.05	0.989264	0	0.989264

Table 5: Finite-law validation of the refinement criterion. All entries are in nats except ϵ. Useful reveals help when their conditional prediction error is sufficiently small; an additional uninformative reveal gives no gain, and a useful reveal with large conditional error ($\epsilon = 0.80$) is worse than one call.

Exhaustive enumeration over all 75 ordered partitions of four tokens at six predictor settings confirms Eq. (6) in every case (450 checks, maximum absolute residual 1.33×10^{-15}) and Eq. (19) over 158 single-group refinements. These analytical results isolate the criterion's two terms; the benchmarks of Section 4 measure the full system.

Formal verification. A Lean 4.24.0 development with pinned mathlib proves the finite nonnegative-target probability construction, the group decomposition, the nested-refinement inequality, and the coverage bound without proof placeholders or additional axioms. True conditional kernels are constructed from a supplied joint target law, with uniform kernels at zero-mass histories; coordinate marginals are computed by summing each group law. Induction then proves normalization and Eq. (6) for any finite sequence of groups with possibly unequal sizes. An explicit coordinate bijection transfers this identity from reveal histories to the original token sequences. Separate theorems establish the normalized feature-attention identity, its nonnegative unit-sum weights under the stated kernel conditions, and the fixed-transition recurrence perturbation bound. The learned categorical kernels in the reveal theorem are strictly positive, which covers stochastic softmax prediction and keeps the forward KL finite.

A.6 Controlled neural evidence at 455M scale

A controlled study with a 455M-parameter bidirectional recurrent denoiser tests the conditional-error mechanism on enumerable binary tasks. This smaller model ties the parameters of its two directions and evaluates its vocabulary head at selected answer positions, whereas the 3.3B model uses separately trained directions and full-position logits; the study isolates the mechanism, and Section 4 evaluates the full system.

Each target in the controlled study is uniform on sixteen valid eight-bit answers, with four dependent pairs. A one-call product sampler discards $4 \log 2$ nats of dependence. Two calls that reveal one bit from each pair before its partner reduce this term to zero: Eq. (19) assigns $\log 2$ nats of recovered information to each partner bit. Two calls that keep pairs intact discard the same dependence as one call. Comparing these two-call policies at the same checkpoint isolates reveal structure while holding the call count fixed. For each of the three distance-balanced models, the gain averaged over near, middle, and far positions is within 1.3×10^{-4} nats of the predicted $4 \log 2$. The largest residual for an individual position's mean is 4.14×10^{-4} nats.

A separate matched training intervention holds context length at 16k and changes task-bearing block positions from near-only to distance-balanced sampling. Three seed-matched adaptation pairs share one selected starting checkpoint. Distance-balanced training lowers far-context conditional error in every adaptation seed. Exhaustive changes to task evidence test whether the learned conditionals respond to that evidence, and a fresh layout panel separates instruction-header position from task-block position (Table 6).

Among the balanced models, the exhaustive probe changes three of 55,296 unaffected-token predictions, so the evidence response is highly but not perfectly selective.

Intervention and metric	Reduction (nats)	95% interval
Training position: far answer KL	6.051	[5.392, 6.735]
Evidence flips: far paired token CE	1.478	[1.327, 1.637]
Nearby header, distant task: answer KL	7.627	[6.908, 8.301]

Table 6: Controlled neural evidence from the 455M study. The first row compares near-only with distance-balanced adaptation; the second and third report the evidence-flip and layout interventions described in the text; intervals bootstrap paired problems.

B GENERATION AND TRAINING ALGORITHMS

Algorithm 1 LaLanD full-canvas generation (default path)

Require: Observed context $\mathbf{x}$, answer budget $N \geq 1$, rounds $K \geq 1$; denoiser f_θ.

$\mathbf{u} \leftarrow [\texttt{MASK}]^N$; $U \leftarrow \{1, \ldots, N\}$; $n \leftarrow 0$.
for $k = 1, \ldots, K$ **do**
  if $U = \varnothing$ **then**
    break
  end if
  Recompute bidirectional features from $[\mathbf{x}; \mathbf{u}]$ with fresh scan states.
  Suppress mask and pad logits; obtain categorical probabilities.
  For $i \in U$, get greedy token $\hat{y}_i$ and confidence c_i.
  if $k = K$ **then**
    $B \leftarrow U$.
  else
    $b \leftarrow \min(|U|, \max(0, \lfloor Nk/K \rfloor - n))$.
    $B \leftarrow$ the b highest-confidence positions in U.
  end if
  Set $u_i \leftarrow \hat{y}_i$ for $i \in B$; record commitment round k.
  $U \leftarrow U \setminus B$; $n \leftarrow n + |B|$.
end for
return $\mathbf{u}$ and commitment rounds.

The procedure also accepts partially observed answer canvases, in which case the commitment budget counts only initially masked positions. If $N < K$, early rounds may commit no tokens and the final round fills the remaining canvas. Mask and pad tokens are excluded from the output vocabulary. Text after the first predicted EOS is discarded.

Algorithm 2 Token-denoiser training step

Require: Pretrained causal RWKV-7 weights; eligible-token masks; corruption recipe and adaptation schedule.

Clone attention into separate forward/reverse modules; initialize gates.
for each minibatch of clean token sequences $\mathbf{s}$ **do**
  Sample a corruption mask $\mathbf{m}$ over eligible positions.
  Replace selected tokens by $[\texttt{MASK}]$ to obtain $\tilde{\mathbf{s}}$.
  Recompute bidirectional logits $f_\theta(\tilde{\mathbf{s}})$.
  Compute masked-token loss using the global masked-token count.
  Compute the forward-only auxiliary causal loss on clean $\mathbf{s}$.
  Backpropagate $\mathcal{L}_{\text{mask}} + \lambda_c \mathcal{L}_{\text{causal}}$.
  Apply the stage-dependent gradient schedule; perform the optimizer step.
end for

Padding and document boundaries. The loss mask selects supervised tokens. The recurrent scan carries state across packed-document boundaries in both directions, as in the causal RWKV-7 pretraining recipe.

C Implementation and evaluation details

C.1 Architecture, initialization, and adaptation recipe

LaLanD has 3.3B parameters, with $L = 32$ layers, width $d = 2560$, 40 heads of dimension 64, and vocabulary size $V = 65{,}536$. Each layer contains two RWKV-7 attention modules, one shared FFN, and normalization layers; both directions use a common token embedding and output head. Cloning the attention modules into two directions and adding the fusion gates ($L(d^2 + d) \approx 0.21$B parameters) accounts for the increase from the 2.9B initialization to 3.3B.

Initialization copies each pretrained attention tensor into both directions; only the fusion gates are new parameters, and their initial bias favors the pretrained forward branch.

Corruption and auxiliary-loss settings. Corruption is sampled per 256-token training window: the masking rate is drawn uniformly from $[0.05, 0.30]$, $[0.30, 0.70]$, or $[0.70, 1.00]$ with probabilities 0.30, 0.40, and 0.30, and a window uses contiguous-span corruption with probability 0.15. Windows define corruption only; every model call scans the full row. In the prefix-completion component, 30% of rows retain a prefix sampled between 5% and 60% of the sequence length and mask the suffix. The adaptation data are 30B tokens resampled solely from the RWKV World v3 corpus that trained the initialization; no external corpus is added. Training runs 3,000 steps on 16 H200 GPUs with learning rate 2×10^{-5}, 200 warm-up steps, and 4,096-token packed rows, starting from the RWKV-7 World3 checkpoint. The auxiliary causal weight λ_c is 0.30 for the first 0.5×10^9 tokens and 0.10 thereafter.

Staged gradients. For the first 5% of steps only the reverse attention and fusion gates receive gradients; from 5% to 15% all parameters in the upper half of the layers are enabled; thereafter all parameters train. Suppressed gradients are zeroed before the AdamW update.

C.2 Model families

Autoregressive references include Qwen2.5, Llama-3.2, Phi-2, Gemma-2, DCLM, OLMo-2, and Pythia (Qwen Team, 2024; Meta, 2024; Microsoft, 2023; Gemma Team et al., 2024; Li et al., 2024; Walsh et al., 2025; Allen Institute for AI, 2025; Biderman et al., 2023). Recurrent/state-space references are GLA, RWKV-6/7, and Mamba-1 (Yang et al., 2024; Peng et al., 2024; 2025; Gu & Dao, 2024). Discrete diffusion references are MDLM and SEDD (Sahoo et al., 2024; Lou et al., 2024). The broader table also includes continuous-representation baselines CoLA-DLM, ELF, and STAR-LDM (Guo et al., 2026; Hu et al., 2026; Lovelace et al., 2025). Model names refer to public configurations; the Total column reports measured parameter counts.

C.3 Capability protocols and controls

The task matrix includes MMLU, OpenBookQA, SocialIQA, Story Cloze, HellaSwag, PIQA, ARC-e/c, SciQ, CommonsenseQA, LAMBADA, RACE, and SQuAD (Hendrycks et al., 2021; Mihaylov et al., 2018; Sap et al., 2019; Mostafazadeh et al., 2016; Zellers et al., 2019; Bisk et al., 2020; Clark et al., 2018; Welbl et al., 2017; Talmor et al., 2019; Paperno et al., 2016; Lai et al., 2017; Rajpurkar et al., 2016). Every cell uses 200 examples, answer generation with a 32-token output budget and seed 42, and exact-match scoring in which an invalid answer counts as incorrect; options are inlined for PIQA, ARC-e/c, SciQ, and CommonsenseQA, and SQuAD reports token F1. Avg_{10} is the unweighted mean of MMLU, OpenBookQA, SocialIQA, Story Cloze, PIQA, ARC-e, SciQ, CommonsenseQA, LAMBADA, and RACE; OLMo lacks some of these tasks, so its entry is omitted. Gemma's leading BOS token is restored in capability runs and suppressed for bAbILong.

C.4 Long-context protocols

RACE scores in the long-context tables reuse the 200-example capability measurement. bAbILong uses official label matching, a fixed evidence/haystack order, and 100 examples per question type and nominal length, except Mamba's 50-example rows. A ‡ marks architecture-limited inputs: Phi-2 receives 2,016 tokens at nominal lengths of 4k or more, and Gemma's positional ceiling limits its 16k input. LongBench is reported for the paired RWKV-7 and LaLanD comparison only.

LongBench uses the official evaluation script on the complete test sets of twenty tasks with the official prompts, generation budgets, and per-task metrics: F1 for NarrativeQA, Qasper, MultiFieldQA-en/zh, HotpotQA, 2WikiMQA, MuSiQue, and TriviaQA; ROUGE-L for DuReader, SAMSum, GovReport, QMSum, MultiNews, and VCSUM; accuracy for TREC, LSHT, PassageCount, and PassageRetrieval-zh; and edit similarity for LCC and RepoBench-P. Both systems are scored on identical inputs, and the 20-task mean is the unweighted average over these tasks.

C.5 Efficiency measurement

Table 3 is measured on one H100 GPU. Every system receives a 512-token prompt and generates a 512-token answer at batch size one, and LaLanD decodes with $K = 16$ rounds. Speed is the number of generated tokens divided by generation time. For autoregressive systems, generation time spans the 512 sequential decoding steps; for LaLanD, it spans the 16 full-canvas rounds and their confidence commits. Total time covers the complete request, from prompt ingestion through the final generated token, and therefore includes all LaLanD planning rounds. Memory is the peak GPU allocation during the request; each reported value is the envelope (maximum) over the measured batches. Prompt and answer lengths are counted in each system's native tokenizer, so an equal token budget corresponds to different amounts of text across systems.

D Complete benchmark results

D.1 Capability matrix

Table 7 shows that Qwen2.5-3B and Phi-2 lead the ten-task aggregates at 70.7% and 68.5%, with LaLanD at 68.3%. LaLanD has the highest SciQ and LAMBADA scores among the displayed systems. Bold and underline rank the best and second-best distinct values among the displayed rows.

Table 7: **Complete short-context results.** Thirteen task columns report accuracy (%), except SQuAD, which reports token F1; Avg_{10} is the ten-task aggregate. LaLanD has 3.3B parameters; Appendix C.3 specifies the evaluation protocol. Higher is better, with bold and underline marking the best and second-best distinct scores.

(a) Knowledge and commonsense

Model	Total	MMLU	OBQA	SIQA	StoryC.	Hella.	PIQA	ARC-e
Autoregressive Transformers								
Qwen2.5-3B	3.1B	**61.0**	73.5	36.0	**90.0**	**57.0**	**79.0**	80.5
Qwen2.5-1.5B	1.5B	58.5	68.5	**44.0**	84.5	43.0	72.5	78.5
Llama-3.2-3B	3.2B	54.0	63.0	39.5	68.0	39.0	65.0	68.0
Phi-2	2.8B	56.0	**76.0**	39.0	85.5	36.5	63.5	**88.0**
Gemma-2-2B	2.6B	49.0	52.5	39.0	59.5	30.5	32.0	65.5
DCLM-1B	1.4B	45.0	50.5	37.0	59.0	34.0	42.0	66.0
Pythia-2.8B	2.9B	30.0	23.5	33.5	51.5	21.0	51.0	27.5
OLMo-2-1B	1.5B	43.0	41.5	39.0	66.0	–	57.5	56.5
Linear-attention and state-space models								
GLA-1.3B	1.4B	10.5	13.0	24.0	26.0	8.5	44.0	31.0
RWKV-7 World3	2.9B	36.5	43.0	20.0	50.0	17.0	61.0	75.5
RWKV-6 World2.1	3.1B	27.0	32.0	35.0	54.0	27.0	48.0	26.0
Mamba-1	2.8B	27.0	16.0	20.0	20.5	19.5	7.5	27.5
Discrete diffusion								
MDLM-OWT	0.2B	3.5	3.5	3.0	7.0	0.5	2.5	0.5
SEDD-medium	0.4B	9.5	6.5	11.0	20.5	3.0	4.5	6.5
Continuous latent and embedding-space models								
CoLA-DLM	2.33B	20.0	23.0	30.0	47.0	23.0	32.5	26.5
ELF-L	0.7B	2.0	1.5	0.5	0.0	0.0	0.5	1.0
ELF-M	0.3B	2.0	1.0	0.5	1.0	0.0	0.5	2.5
STAR-LDM	2.36B	8.0	10.0	11.5	2.0	9.5	14.0	11.5
Bidirectional recurrent diffusion								
LaLanD	3.3B	**61.0**	62.5	41.0	78.0	41.5	66.5	78.0

Table 7 (continued). Complete short-context results; the same systems and ranking conventions as panel (a).

(b) Reasoning, completion, and extraction

Model	ARC-c	SciQ	CQA	LAMB.	RACE	SQuAD	Avg_{10}
Autoregressive Transformers							
Qwen2.5-3B	76.5	83.5	**67.5**	59.5	**76.5**	84.4	**70.7**
Qwen2.5-1.5B	70.5	81.0	61.5	56.0	74.5	78.0	68.0
Llama-3.2-3B	64.0	73.5	51.0	64.0	62.5	84.8	60.9
Phi-2	**81.0**	86.0	64.0	61.0	66.0	**87.3**	68.5
Gemma-2-2B	48.0	75.5	47.0	53.5	53.5	76.4	52.7
DCLM-1B	47.0	71.0	46.5	64.0	51.5	71.6	53.3
Pythia-2.8B	19.5	25.5	15.5	62.0	20.0	59.0	34.0
OLMo-2-1B	–	67.5	35.5	67.0	–	–	–
Linear-attention and state-space models							
GLA-1.3B	23.0	25.5	17.0	45.5	19.0	8.9	25.6
RWKV-7 World3	64.0	72.0	39.5	76.0	55.0	7.8	52.9
RWKV-6 World2.1	24.5	22.0	20.0	73.0	25.0	44.4	36.2
Mamba-1	23.0	33.5	21.5	43.0	24.0	38.7	24.1
Discrete diffusion							
MDLM-OWT	0.0	0.5	2.5	20.5	1.5	4.0	4.5
SEDD-medium	2.5	7.0	5.5	27.5	0.0	12.4	9.9
Continuous latent and embedding-space models							
CoLA-DLM	21.0	23.0	17.5	50.5	20.5	50.3	29.1
ELF-L	0.5	4.5	3.0	1.0	0.0	0.0	1.4
ELF-M	0.5	4.0	4.5	7.5	0.0	0.4	2.4
STAR-LDM	4.0	14.0	9.5	19.5	1.0	2.2	10.1
Bidirectional recurrent diffusion							
LaLanD	60.0	**88.0**	61.0	**79.0**	68.0	26.7	68.3

D.2 Reading and retrieval

Table 8 extends Table 2 with the Transformer references under the positional limits noted in Appendix C.4; LongBench task-level scores are in Table 1.

Table 8: **Long-context reading and retrieval.** RACE and bAbILong report answer accuracy (%). LaLanD has 3.3B parameters. Higher is better; bold and underline mark the best and second-best distinct scores among the rows shown. ‡ marks truncated, architecture-limited inputs (Appendix C.4).

Model	Total	RACE	QA1 4k	QA1 8k	QA1 16k	QA2 4k
Autoregressive Transformers						
Qwen2.5-3B	3.1B	**76.5**	**67.0**	**66.0**	**65.0**	21.0
Llama-3.2-3B	3.2B	62.5	60.0	50.0	44.0	**26.0**
Phi-2	2.8B	66.0	44.0‡	25.0‡	22.0‡	19.0‡
Gemma-2-2B	2.6B	53.5	13.0	3.0	0.0‡	3.0
Linear-attention and state-space models						
GLA-1.3B	1.4B	19.0	14.0	4.0	2.0	7.0
Mamba-1	2.8B	24.0	2.0	2.0	0.0	6.0
RWKV-7 World3	2.9B	55.0	41.0	21.0	17.0	19.0
Bidirectional recurrent diffusion						
LaLanD	3.3B	68.0	44.0	19.0	19.0	20.0