ROLE-CONDITIONED RESIDUAL COMPUTATION IN DIFFUSION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Text diffusion is emerging as a competitive alternative to autoregressive language modeling: it reconstructs masked tokens over successive steps and can predict many positions in parallel. Yet standard Transformer backbones carry visible evidence and evolving masked predictions through a single residual state at each position. We ask whether learned routes through depth can better support denoising. We add persistent residual streams with token-conditioned read, transport, and write maps around unchanged Transformer branches. In matched 337M-parameter pretraining on 3.5B tokens, two streams capture 86.6% of the best additional in-domain gain beyond learned single-stream control and achieve the strongest mean transfer across four external corpora. Without assigned stream roles, visible and masked positions learn different read and write routes. Causal interventions establish the importance of these routes and of token-specific write strength; a role-factorized model trained from scratch closes 50.2% of the gap from one stream to the full two-stream model. Stream-state interventions reveal a preservation–transfer–reuse mechanism in the two-stream model: one stream retains an input-dependent early state, the other carries visible-source influence to masked targets, and late layers reuse the retained state. The preservation-and-reuse signature recurs across diffusion widths and is nearly absent in autoregressive controls. We hope these findings establish multi-stream residual transport as both a practical architecture for text diffusion and a useful lens for understanding how denoising models organize computation between evidence and prediction.

1 INTRODUCTION

Diffusion language models have recently emerged as a compelling alternative to autoregressive language modeling (Austin et al., 2021; Hoogeboom et al., 2021; Sahoo et al., 2024; Nie et al., 2025; Ye et al., 2025). Rather than generating text token by token, they predict many missing positions in parallel and refine uncertain predictions over successive steps. Recent advances in objectives, scaling, and decoding have made this approach increasingly competitive.

The contrast runs deeper than the generation procedure: autoregression and masked diffusion ask token positions to do different kinds of work. Autoregressive positions follow a broadly shared pattern: each processes its available prefix to support next-token prediction. In masked diffusion, visible positions provide observed evidence, while masked positions must construct predictions from that evidence. These roles coexist in every partially corrupted sequence and change as tokens are revealed.

Despite these different roles, most masked diffusion models inherit the standard Transformer residual pathway. Attention and MLP branches can respond to the corruption mask, but at each position they read from and write to a single persistent state. Progress in diffusion has largely changed the objective, corruption process, and decoding procedure while leaving this route through depth intact. This raises the question of whether a shared residual route is the best way to carry observed evidence and evolving predictions through the network.

Work beyond text diffusion shows that residual transport can be made learnable. ReZero and DenseFormer regulate updates or combine information across depth, while Hyper-Connections and mHC introduce multiple persistent streams with learned read, transport, and write patterns (Bachlechner et al., 2021; Pagliardini et al., 2024; Zhu et al., 2025; Xie et al., 2025). In autoregressive

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.

language modeling, positions share the same left-to-right prediction task. Masked diffusion adds an explicit source of heterogeneity: the corruption pattern separates positions that provide evidence from those that must reconstruct missing content. It may therefore organize how a learned residual pathway is used.

We start with a single residual stream and let the strength of each update adapt to the token state. This alone improves diffusion modeling. We then add persistent streams, allowing each branch to choose what it reads, how existing states are carried forward, and where it writes new information. The attention and MLP branches remain unchanged, and no stream is assigned to masked or visible positions. We use interventions on the learned routes and stream states to trace what the model discovers.

Adding a second stream captures most of the gain beyond learned single-stream control. Its computation unfolds through depth: the corruption mask orients where positions read and write, while token-specific write strength matters most for visible evidence early and masked targets later. In the two-stream model, one stream holds a state built in the first block as the other makes visible-source influence available at masked targets. Late layers bring the held state back into computation. The preservation-and-reuse signature recurs across diffusion widths and is nearly absent in autoregressive controls, showing how denoising organizes the residual pathway itself.

We hope these findings motivate diffusion architectures whose information pathways are designed around denoising.

2 Related Work

Text diffusion language models. Discrete diffusion brought corruption and denoising to categorical sequences (Austin et al., 2021; Hoogeboom et al., 2021; Lou et al., 2024); masked diffusion developed the approach for language modeling (Sahoo et al., 2024). Recent systems improve objectives, sampling, and scale (Nie et al., 2025; Ye et al., 2025). We study the residual pathway that their Transformer backbones generally retain.

Mask-dependent computation in diffusion models. Mask tokens affect how MDLMs use visible context (Piskorz et al., 2026); masked and visible positions have different activation distributions (Yan et al., 2026); and their hidden states evolve differently as tokens are revealed (Miele et al., 2026). These observations establish corruption state as a computational variable. We ask whether it also organizes residual routes and the states they carry through depth.

Residual architectures outside text diffusion. ReZero and DeepNorm regulate residual update strength (Bachlechner et al., 2021; Wang et al., 2022); DenseFormer and Attention Residuals select representations across depth (Pagliardini et al., 2024; Kimi Team, 2026). Hyper-Connections instead add persistent streams and learn how branches read, mix, and update them (Zhu et al., 2025). We use mHC-lite to constrain stream mixing exactly with low overhead (Xie et al., 2025; Yang & Gao, 2026). To our knowledge, we are the first to test learned multi-stream residual transport in text diffusion and trace how denoising uses the resulting routes.

Mechanistic analyses of multi-stream residual pathways. Autoregressive HC and mHC analyses find asymmetric stream reliance, concentrated routing, and selective read/write behavior (Peng et al., 2026; Alimaskina et al., 2026; Zhao et al., 2026). Masked diffusion offers a sharper test of organization: its corruption pattern creates explicit, changing roles within the same sequence.

Our contributions are threefold:

- We introduce learned residual transport into masked diffusion while leaving the Transformer branches unchanged. Token-adaptive updates already improve a single-stream model, and two persistent streams capture nearly all of the further gain from increasing residual width.
- Through controlled interventions, we identify the corruption mask as a principal organizer of residual routing. It largely determines the directions in which masked and visible positions read and write, while token-specific state modulates the volume of the resulting updates.

- In the two-stream model, we uncover a preservation–transfer–reuse mechanism: one stream retains an early state while the other carries visible-source influence into masked targets before late reuse. Its preservation-and-reuse signature recurs across diffusion widths and is nearly absent in autoregressive controls.

3 Decomposing Residual Transport

Multi-stream residual interface. Following the mHC notation, let $\boldsymbol{x}_\ell \in \mathbb{R}^{n\times d}$ contain the residual streams of one token entering sublayer ℓ. The interface updates them as

$$\boldsymbol{x}_{\ell+1} = \mathcal{H}_\ell^{\text{res}}\boldsymbol{x}_\ell + \mathcal{H}_\ell^{\text{post}\,\top}\mathcal{F}_\ell(\mathcal{H}_\ell^{\text{pre}}\boldsymbol{x}_\ell)\,. \tag{1}$$

The token-conditioned mappings $\mathcal{H}_\ell^{\text{pre}}$, $\mathcal{H}_\ell^{\text{res}}$, and $\mathcal{H}_\ell^{\text{post}}$ respectively determine what the sublayer reads, how existing streams are carried forward, and how its output is written back. In mHC-lite, $\mathcal{H}_\ell^{\text{res}}$ is exactly doubly stochastic and therefore preserves the mean across streams. Streams are initialized by replication and averaged before the final output head.

Separating write destination from volume. Let $\boldsymbol{g}_\ell = \mathcal{H}_\ell^{\text{post}}$ be the physical write gains. Their total, mean update coefficient, and descriptive write shares are

$$V_\ell = \boldsymbol{g}_\ell \mathbf{1}_n, \qquad v_\ell = V_\ell/n, \qquad p_{\ell i} = g_{\ell i}/V_\ell, \qquad g_{\ell i} = n v_\ell p_{\ell i}. \tag{2}$$

Thus $p_{\ell i}$ describes the fraction of an individual token's update written to stream i, while v_ℓ measures its overall *write volume*. Causal replacements use a bounded destination coordinate so that changed gains remain valid. It agrees with p when the total gain is at most two; Appendix A gives the exact reconstruction.

Because $\mathcal{H}_\ell^{\text{res}}$ is doubly stochastic, it transports the existing streams without changing their mean. Averaging Equation 1 over streams therefore leaves the new sublayer output scaled by v_ℓ. In this precise sense, v_ℓ is a token-dependent residual-update coefficient – the learned counterpart of the unit coefficient in a standard residual connection. The write shares determine which stream states are updated and remain available to later sublayers.

Prior HC-family work parameterizes and analyzes $\mathcal{H}_\ell^{\text{post}}$ jointly, or examines its entries directly, without separating normalized allocation from overall write strength (Zhu et al., 2025; Xie et al., 2025; Yang & Gao, 2026). Write volume is also load-bearing in autoregressive models (Appendix E.1).

Takeaway. Write volume controls how much new computation enters the mean state; destination controls how that update is allocated across persistent streams.

4 How Many Residual Streams Does Diffusion Need?

Before analyzing how the streams are used, we determine how many are useful. We train masked-diffusion models with $n = 1, 2, 3, 4$ on the same ordered 3.5B-token sample from FineWeb-Edu (Penedo et al., 2024). The learned-interface sweep shares a 16-block Qwen-style backbone (Qwen Team, 2025), Dream tokenizer (Ye et al., 2025), and optimizer grouping; the residual interface varies with width. The single-stream variant learns token-adaptive updates, while wider variants add persistent states. A separately trained standard-residual model supplies a contextual reference. Appendix B gives full details.

Adding a second state reduces validation CE by 0.0335 beyond learned $n = 1$, capturing 86.6% of the largest additional gain in the matched

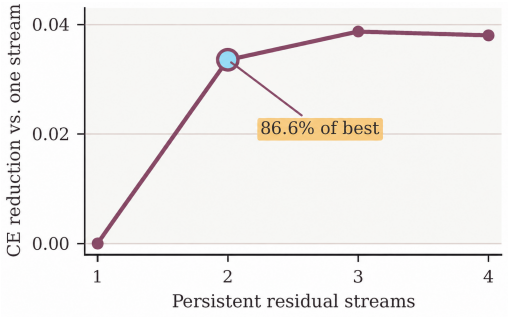

Figure 1: **Two streams capture most width gain.** Validation-CE reduction relative to learned $n = 1$. The $n = 2$ model captures 86.6% of the largest additional gain in the matched width sweep. Blue marks the analyzed model.

Table 1: **Two streams transfer best.** Relative reduction in exponential masked-token CE (perplexity, %) versus the standard-residual reference. Higher is better.

Interface	OWT	WT103	LAMBADA	C4	**Mean**
Std. residual	0.00	0.00	0.00	0.00	0.00
$n = 1$	1.44	1.17	-0.15	1.77	1.06
$n = 2$	4.61	**7.61**	**3.02**	5.29	**5.13**
$n = 3$	**5.51**	6.15	2.27	5.17	4.78
$n = 4$	5.14	3.30	2.67	**5.39**	4.13

width sweep (Figure 1). Increasing width to $n = 3$ or $n = 4$ reduces CE by just a further 0.0052 or 0.0045. We select $n = 2$ as the smallest model delivering nearly all the benefit of additional streams.

To test whether the choice of $n = 2$ transfers beyond FineWeb-Edu, we evaluate all checkpoints on OpenWebText, WikiText-103, LAMBADA, and C4 using matched examples and masks at three corruption levels. Table 1 reports relative reductions in exponential masked-token CE; $n = 2$ leads the mean across the learned-interface sweep. Full details and paired bootstrap intervals appear in Appendix B.

Takeaway. The second persistent stream delivers 86.6% of the available gain beyond learned single-stream control, and $n = 2$ leads mean transfer across four corpora. It is the clearest setting for tracing how denoising uses residual transport.

5 The Corruption Mask Organizes Residual Routing

The width sweep shows that a second state helps, but not how it is used. Routing could depend on each token's full state, or it could follow the masked/visible role supplied by the corruption pattern. We distinguish these possibilities by replacing dynamic controls with averages. If the mask organizes routing, separate averages for masked and visible positions should preserve performance, while swapping them should hurt.

Causal interventions. We study the $n = 2$ model at $t = 0.5$, where masked and visible positions are both well represented. For each of four controls – read, transport, write destination, and write volume – we replace its dynamic value throughout all attention and MLP interfaces while leaving the other three coordinates dynamic. A *global mean* removes token dependence; *role means* retain only the masked/visible distinction; and a *role swap* exchanges the two role-conditioned values. We also replace all four jointly. Means are estimated on disjoint calibration sequences using only the observed corruption mask.

Figure 2 shows that collapsing read or write destination to a single mean substantially degrades performance. In contrast, averaging within masked and visible positions separately nearly preserves both controls. Swapping their means again causes a large degradation, showing that the two settings are specifically assigned to visible evidence and masked prediction targets. Stream mixing behaves differently: its global mean is already nearly sufficient, suggesting that its average setting at each layer matters more than token-specific variation. These results suggest a simple factorization in which token role determines the main routing decisions.

Training a role-factorized interface. The interventions describe how a fully dynamic model behaves after training. We next test whether the same factorization can guide learning from scratch. We therefore train a new two-stream model, FRR, whose interface follows the pattern above. Masked and visible positions have separate learned read and write-destination settings, while a scalar predictor retains token-dependent write volume. Stream transport is fixed to the identity. FRR is trained from initialization under the same data order, masks, backbone architecture, and optimization schedule as the other learned-interface models (Appendix C).

FRR reaches a validation CE of 3.5634, improving on learned $n = 1$ (3.5802) and closing 50.2% of the gap to full $n = 2$ (3.5467). Relative to the standard-residual reference (3.6055), it retains 71.6%

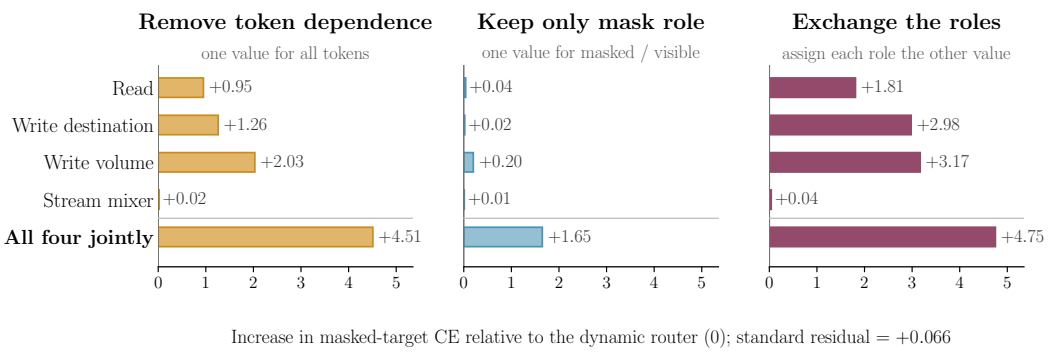

Figure 2: **Masked and visible tokens use distinct residual routes.** Role-conditioned means nearly recover the dynamic read and write destination, whereas exchanging the two roles performs worse than the standard residual model. Stream mixing is largely shared, while write volume retains substantial within-role dependence. Lower masked-target CE is better. Full estimates and confidence intervals appear in Appendix D.

of the observed full-model gain. The trained result shows that two persistent states already help when their basic read and write directions are learned from masked/visible role and write strength remains token-dependent.

Role and state jointly organize the interface. Figure 2 sharpens this factorization. Role-specific means nearly preserve read and write destination, whereas write volume still varies consequentially among tokens with the same role. FRR turns that finding into a trainable two-stream interface: role-specific read and write directions are paired with token-dependent scalar volume, and stream transport is fixed to the identity. The full dynamic interface improves further. Together, the trained comparison and causal interventions establish both the practical value of role factorization and the importance of token-specific write strength.

Takeaway. The corruption mask acts as a routing signal: visible evidence and masked targets follow different read and write paths, while token state adjusts update strength within each path. FRR shows that this division is useful when learned from scratch.

6 Write Volume Schedules Denoising Through Depth

The previous section shows that the corruption mask largely determines the route taken by an update. It does not fully determine write strength: volume v_ℓ controls the overall update, while destination allocates it across streams. Positions can therefore use similar routes while modifying their states by very different amounts.

This distinction is clearly visible in the role-mean interventions. At $t = 0.5$, replacing volume with its masked/visible mean increases CE by $+0.200$, compared with $+0.041$ for read, $+0.017$ for write destination, and $+0.006$ for transport. Write volume thus carries substantial information beyond token role. We next ask what this variation represents: whether it is merely local rescaling, or whether it organizes denoising across tokens and depth.

Causal interventions. We address this question with three complementary interventions. First, we test whether each volume must remain attached to the token state that produced it. A *role-mean replacement* assigns every selected position the calibrated mean for its role and interface, removing all within-role variation. A *within-role shuffle* instead preserves the exact set of volume values but reassigns them among positions of the same role. It keeps the role-specific distribution while breaking the correspondence between tokens and write strengths. Both interventions preserve each affected token's current bounded destination coordinate as its volume changes.

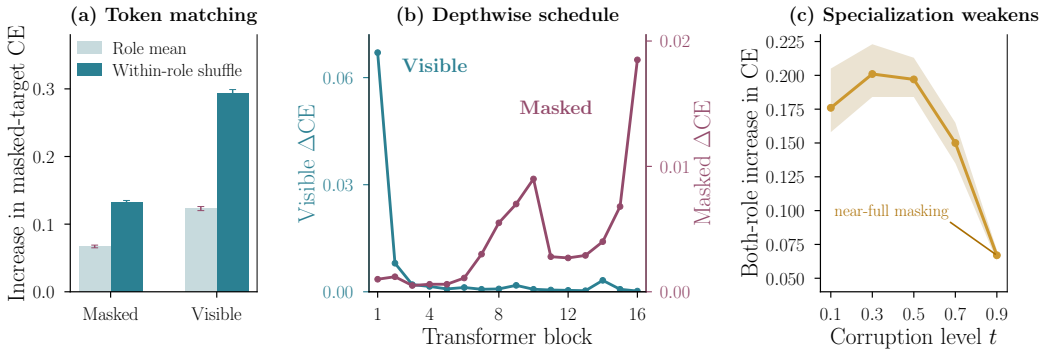

Figure 3: **The functional role of token-specific write volume shifts through depth.** (a) At $t = 0.5$, within-role shuffling is more damaging than role-mean replacement despite preserving the volume distribution. (b) Single-block interventions reveal a handoff from visible positions early to masked positions late; effects use separate zero-anchored axes. (c) Dependence on token-specific volume weakens near full masking. Error bars and bands show paired 95% sequence-bootstrap intervals; $n = 1{,}024$ in (a–b) and 128 per corruption level in (c).

We then locate this dependence through depth by applying the role-mean replacement to one block at a time. Finally, we repeat the full-network replacement across corruption levels to determine when token-specific volume is most useful. Read and transport remain dynamic; calibration uses disjoint sequences, and effects are measured by the increase in masked-target CE.

Write strength is matched to token state. We begin by testing whether the assignment of volume values matters. Within-role shuffling is more destructive than role-mean replacement for both roles (Figure 3a). For masked positions, shuffling increases CE by $+0.132$, compared with $+0.068$ under averaging; for visible positions, the corresponding effects are $+0.294$ and $+0.124$. Thus, preserving the full distribution is not enough. Indeed, assigning plausible volume values to the wrong tokens is more harmful than removing within-role variation altogether. Write strength must therefore remain matched to the token state that produced it. The larger effect for visible positions further suggests that the model is especially selective about which pieces of observed evidence receive strong updates.

Control shifts from evidence to prediction through depth. Having established that token–volume matching matters, we next ask where it is used. The single-block interventions reveal a clear handoff between the two roles (Figure 3b). In the first block, replacing visible-token volume increases CE by $+0.0673$, while replacing masked-token volume has an effect of only $+0.0015$. Visible sensitivity then declines through depth, whereas masked sensitivity grows and peaks near the readout.

Write volume thus serves distinct functions at different stages of the network. Early layers regulate how strongly visible evidence enters the computation. As representations develop, this control shifts toward masked positions, where it governs the strength of prediction updates. The resulting schedule moves from evidence integration to prediction refinement within a single denoising pass.

Token-specific scheduling is strongest under partial observation. This interpretation predicts that token-specific volume should matter most when visible evidence and masked targets must be coordinated. The corruption sweep supports this prediction (Figure 3c). Replacing volume with role means increases CE by $+0.177$–$+0.201$ for $t = 0.1$–0.5, where both roles are present in substantial numbers. At $t = 0.9$, almost every position is masked, little visible evidence remains, and the effect falls to $+0.067$. The need for token-specific control therefore weakens as denoising becomes less dependent on interactions between observed and missing content.

Takeaway. Within each role, write strength must stay matched to the token. The model uses it to weight visible evidence early and masked prediction later, with the strongest effect when observed and missing content coexist.

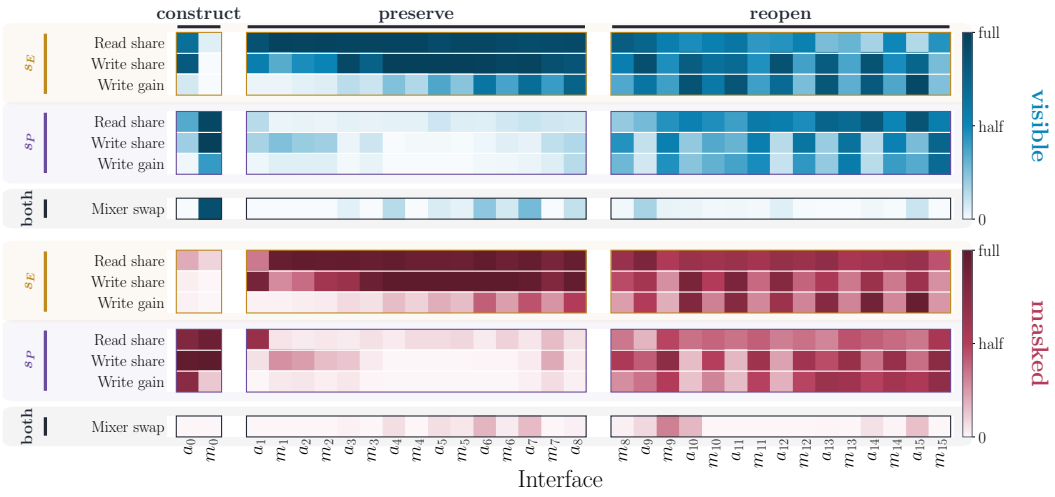

Figure 4: **Routing closes the preserved stream through middle depth and reopens it late.** Columns are the 32 interfaces, attention a_k and MLP m_k per block, with gaps at the phase boundaries. Rows are grouped by stream: read share, write share from role-mean gains $\overline{g}_{\ell i}/\sum_j \overline{g}_{\ell j}$, and mean physical write gain $\overline{g}_{\ell i}$ for s_{E} and for s_{P}; mixer off-diagonal mass belongs to the interface and is shown once. Panels are position roles, visible above, masked below. Colour is normalized within each row – 0–1 for the shares, 0–2 for the gain, 0–0.25 for the mixer mass – so shades compare along a row, not between rows. Across a_1–a_8 the preserved stream takes 6% of the read mass and 3% of the write gain; late layers reopen it.

7 Residual Streams Separate State Preservation from Evidence Transfer

The preceding analyses show that the corruption mask orients where updates are routed, while write volume controls how strongly individual positions modify the residual state. We now ask how the model divides computation between its two streams. Figure 4 reveals a staged pattern. In the opening block, the model constructs a state that depends on whether a position is masked or visible. Through the middle layers, it preserves this state almost unchanged in one stream, while the other transfers information from visible tokens to masked targets. Near the readout, the preserved state returns to the computation. The two streams let the model integrate evidence while retaining an early state for later use.

We therefore refer to stream 0 as the *evolving stream*, s_{E}, and stream 1 as the *preserved stream*, s_{P}. These names describe their roles through the middle of the network: s_{E} carries the ongoing computation, while s_{P} retains the state formed near the input.

The first block determines the state to preserve. The construction columns of Figure 4 show that the streams begin identically but separate immediately through role-dependent attention routing:

$$(s_{\mathrm{E}}, s_{\mathrm{P}})_{\text{after } a_0} \approx \begin{cases} (x + \Delta a_0,\ x), & \text{visible,} \\ (x,\ x + \Delta a_0), & \text{masked.} \end{cases}$$

The corresponding destination weights are $(0.88, 0.12)$ and $(0.02, 0.98)$, respectively, and the masked attention update is substantially larger than the visible update. When the input token is observed, the first attention branch therefore updates s_{E} while leaving s_{P} close to the input state. When the token is masked, it instead places a large context-derived update into s_{P}. Because the following MLP reads mainly from s_{P}, it receives an observed-token-derived state at visible positions and a strongly context-updated state at masked positions. After m_0, this role-dependent state enters the preservation regime.

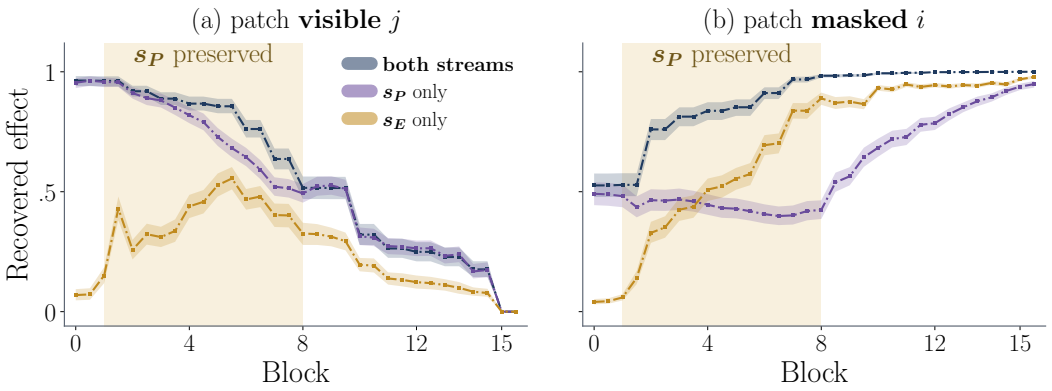

Figure 5: **Visible-source influence reaches masked targets during preservation.** Restoring both streams at the visible source (left) repairs progressively less of its ablation effect, while restoring s_{E} at the masked target (right) repairs progressively more. Mean recovery and 95% bootstrap confidence intervals over 179 pairs; shading marks the s_{P}-preservation interval.

This routing choice is causal. Directing the masked attention update to the stream subsequently read by the MLP recovers 98.2% of its effect. At visible positions, sending the update to the opposite stream increases CE by 3.04, more than removing it altogether. The first block therefore constructs different early states for evidence and prediction rather than merely separating two identical copies.

The middle network preserves a position-specific early state. In the preservation region of Figure 4, attention and MLP branches predominantly read from s_{E} and write back to it. Read and write mass on s_{P} become small, while the bottom row shows that the mixer remains nearly diagonal. The preserved state is therefore neither continually rewritten nor silently transferred into the evolving stream. State similarity provides complementary evidence: s_{P} remains close to its post-m_0 state throughout this interval (Appendix G).

Two interventions distinguish preservation from disuse. For the same token, replacing s_{P}^{a8} with its post-m_0 state changes masked-target CE by just $+0.006$ when patching visible positions and $+0.004$ when patching masked positions. Shuffling s_{P} among same-role positions within each sequence is destructive; restoring each position's matched post-m_0 state removes 95% and 98% of the damage for masked and visible positions, respectively (Appendix G). Thus, the early state remains useful at its original position even while changing little across middle depth. The evolving stream absorbs ongoing updates; we next ask what they accomplish.

Where visible evidence becomes available. The middle network largely leaves s_{P} untouched. To determine what happens to visible evidence during this interval, we mask a visible source j and measure the resulting loss increase at a masked target i. After each block, we restore the clean s_{E} state, s_{P} state, or both at either position. Recovery is the fraction of the source-ablation effect removed by the patch. Since MLPs act independently at each position, source-to-target influence must pass through attention.

Figure 5 shows that the transfer takes place within the shaded preservation interval, before s_{P} reopens. At its start, restoring the visible source repairs nearly all of the ablation effect, while restoring s_{E} at the masked target repairs almost none. Across the middle blocks, recovery from patching even both source streams falls, whereas recovery from the target's s_{E} rises to nearly complete. Throughout this transition, recovery from the target's s_{P} remains roughly flat. These blocks therefore perform the evidence transfer: attention makes source influence available in the masked target's evolving stream while its early s_{P} state is held aside. Only later does recovery from the target's s_{P} rise, as that stream reopens.

Late blocks return the preserved state to computation. The reopening region of Figure 4 begins around m_8–m_{10}: branches again read from and write to s_{P}. Concurrently, its similarity to the post-m_0 state declines (Appendix G), indicating that preservation gives way to transformation.

Late reuse remains role-dependent. After m_{12}, collapsing the visible streams to s_{P} is nearly harmless ($\Delta\mathrm{CE} = +0.010$), whereas collapsing them to s_{E} costs $+0.148$. At masked positions, collapsing to s_{E} or s_{P} costs $+0.077$ and $+0.230$, respectively, and removing s_{P} before the head costs $+0.355$. Visible positions therefore return primarily to the retained state, whereas masked predictions continue to depend on both the evolving computation and the reactivated early state. The preserved stream is not an unused bypass: it is held aside through evidence transfer and brought back before readout.

The preservation-and-reuse signature is nearly absent in AR. To test whether preservation and late reuse are specific to one checkpoint, we survey routing and early-state similarity across diffusion widths $n \in \{2, 3, 4\}$ and both token roles. Every diffusion model contains a stream with little middle-depth reading, high similarity to its early state, and stronger late reading. The matched AR checkpoints do not reproduce this joint signature: their candidate streams are read more during middle depth and depart sooner from the early state; the $n = 2$ AR candidate is read throughout this interval. Thus the preservation-and-reuse organization recurs across diffusion widths and is nearly absent in AR (Appendix H). The source-to-target transfer intervention above establishes the complementary evidence-flow step in the two-stream diffusion model.

Takeaway. In the two-stream model, an input-dependent first-block state remains available while visible-source influence reaches masked targets. Late layers reuse that state; the preservation-and-reuse signature extends across diffusion widths.

8 Discussion

Masked diffusion gives visible and masked positions complementary roles within each denoising pass. The learned residual interface reflects this structure: the corruption mask orients read and write routes, while token state controls write strength. In the full two-stream model, an early position-matched state persists through the middle network as the evolving stream carries visible-source influence to masked targets; late layers reuse the preserved state. This preservation-and-reuse signature recurs across diffusion widths and is nearly absent in matched autoregressive controls, connecting the architectural gain to a distinctive organization of denoising computation.

The mechanism also guides architecture design. A role-factorized interface trained from scratch closes 50.2% of the gap from a learned single stream to the full two-stream model. Its performance shows that role-conditioned routes paired with token-dependent write strength are useful design components; the full dynamic interface improves further. These results turn the observed routing pattern into a concrete direction for diffusion backbones.

9 Conclusion

We introduced learned residual transport into text diffusion and showed that two streams capture 86.6% of the largest additional validation-CE gain beyond a learned single stream while achieving the strongest mean transfer across four external corpora. Causal interventions reveal how the streams preserve an early state, transfer visible evidence to masked targets, and reuse the preserved state near readout. Together, the architectural results and mechanistic evidence establish residual transport as a design axis for diffusion language models and lay the groundwork for architectures built around denoising-specific information flow.

Reproducibility Statement

Appendix A specifies architectures, data order, token budgets, mask construction, intervention reconstruction, calibration/test separation, and bootstrap procedure. All reported intervention deltas use identical held-out sequences within a comparison. Code is available anonymously at `https://anonymous.4open.science/r/residual_diffusion-ED97/`.

ETHICS STATEMENT

This work studies internal computation in language models and introduces no new data collection involving people. The training corpora inherit the biases and licensing constraints of their source datasets. Better understanding of diffusion-model internals can support auditing, but may also improve systems that reproduce harmful content; deployment requires the same evaluation and safeguards as other language models.

AI USE STATEMENT

Generative AI tools were used for editorial assistance, code review, and figure-layout iteration. All claims, equations, experiment definitions, numerical results, and citations were checked against the underlying experiment artifacts by the authors.

REFERENCES

Ekaterina Alimaskina, Gleb Molodtsov, and Aleksandr Beznosikov. Analyzing stream collapse in hyper-connections: From diagnosis to mitigation. *arXiv preprint arXiv:2606.03483*, 2026. URL https://arxiv.org/abs/2606.03483.

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://arxiv.org/abs/2107.03006.

Thomas Bachlechner, Bodhisattwa Prasad Majumder, Henry Mao, Gary Cottrell, and Julian McAuley. ReZero is all you need: Fast convergence at large depth. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 1352–1361. PMLR, 2021. URL https://proceedings.mlr.press/v161/bachlechner21a.html.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Advances in Neural Information Processing Systems*, volume 34, pp. 12454–12465, 2021. URL https://arxiv.org/abs/2102.05379.

Kimi Team. Attention residuals. *arXiv preprint arXiv:2603.15031*, 2026. URL https://arxiv.org/abs/2603.15031.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2310.16834.

Andrea Miele, Yiming Qin, Alba Carballo-Castro, Justin Deschenaux, and Pascal Frossard. Fixed-point masked generative modeling, 2026. URL https://arxiv.org/abs/2605.31215.

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, 2025. URL https://arxiv.org/abs/2502.09992.

Matteo Pagliardini, Amirkeivan Mohtashami, François Fleuret, and Martin Jaggi. DenseFormer: Enhancing information flow in transformers via depth weighted averaging. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2402.02622.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2406.17557.

William Peng, Josheev Rai, Kevin Tseng, Siwei Wang, and Sean Wu. Ablate and rescue: A causal analysis of residual stream hyper-connections, 2026. URL https://arxiv.org/abs/2603.14833.

Julianna Piskorz, Cristina Pinneri, Alvaro Correia, Motasem Alfarra, Risheek Garrepalli, and Christos Louizos. Masks can be distracting: On context comprehension in diffusion language models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2511.21338.

Qwen Team. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

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. URL https://arxiv.org/abs/2406.07524.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. DeepNet: Scaling transformers to 1,000 layers. *arXiv preprint arXiv:2203.00555*, 2022. URL https://arxiv.org/abs/2203.00555.

Zhenda Xie, Yixuan Wei, Huanqi Cao, Chenggang Zhao, Chengqi Deng, Jiashi Li, Damai Dai, Huazuo Gao, Jiang Chang, Kuai Yu, Liang Zhao, Shangyan Zhou, Zhean Xu, Zhengyan Zhang, Wangding Zeng, Shengding Hu, Yuqing Wang, Jingyang Yuan, Lean Wang, and Wenfeng Liang. mHC: Manifold-constrained hyper-connections, 2025. URL https://arxiv.org/abs/2512.24880.

Xin Yan, Aqiang Wang, Zhenglin Wan, Xingrui Yu, and Ivor W. Tsang. STaR-Quant: State-time consistent post-training quantization for diffusion large language models, 2026. URL https://arxiv.org/abs/2606.04945.

Yongyi Yang and Jianyang Gao. mHC-lite: You don't need 20 Sinkhorn–Knopp iterations, 2026. URL https://arxiv.org/abs/2601.05732.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models, 2025. URL https://arxiv.org/abs/2508.15487.

Pengxiang Zhao, Xing Li, Xianzhi Yu, Wei Guo, and Zhenhua Dong. How does mHC use its residual streams? selective routing and near-identity mixing. *arXiv preprint arXiv:2609.05309*, 2026. URL https://arxiv.org/abs/2609.05309.

Defa Zhu, Hongzhi Huang, Zihao Huang, Yutao Zeng, Yunyao Mao, Banggu Wu, Qiyang Min, and Xun Zhou. Hyper-connections. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2409.19606.

A EXPERIMENTAL PROTOCOL AND RESIDUAL COORDINATES

Pretraining and validation. All width and FRR comparisons use a 16-block Qwen-style Transformer with 1,024 hidden dimensions, 2,816 MLP dimensions, 16 query heads, four key–value heads, and tied embeddings for the 152,064-token Dream vocabulary (tokenizer revision `ebb8da7f`). The ordered training sample is from FineWeb-Edu 10BT (revision `87f09149`). Each run uses 1,024-token sequences, 53,400 optimizer updates, and 65,536 effective tokens per update: 3,499,622,400 nominal training tokens. Each block contains one attention and one MLP residual interface, giving 32 routing sites. The mechanistic analyses use the final $n = 2$ checkpoint at update 53,399.

At training time, each sequence draws a corruption level $t \sim \mathrm{Uniform}(0, 1)$, floored at 0.02. Tokens are masked independently with probability t, with one position forced masked if the draw produces none. The MDLM objective averages the masked token CE sum divided by $1{,}024t$ across sequences. This is the same corruption and normalization rule for all MDLM interfaces.

The in-domain validation comparison uses the same 64 held-out blocks and fixed masks for each model. Its MDLM score is the equally weighted mean of masked-reconstruction CE at mask ratios $r \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, with each ratio normalized by the historical expected masked count $1{,}024r$. AR validation uses the $64 \times 1{,}023$ next-token targets. Within each objective, the same validation protocol is applied at the same training step.

External-corpus evaluation. OpenWebText, WikiText-103, LAMBADA, and C4 each contribute 256 token blocks. All models receive the same blocks and masks at $r \in \{0.1, 0.5, 0.9\}$, with mask seeds 901, 905, and 909. We calculate actual-masked-token CE separately at each ratio and average the three CE values. The relative improvement shown in the main text is $100[1 - \exp(\mathrm{CE}_{\mathrm{model}} - \mathrm{CE}_{\mathrm{residual}})]$. Paired intervals use 10,000 sequence resamples, shared across models and ratios.

Causal interventions. Unless stated otherwise, calibration means are computed in float64 on 64 disjoint sequences, separately for each of the 32 interfaces. The role and volume interventions use 1,024 common held-out sequences at $r = 0.5$. Every condition uses the same corrupted inputs and masked targets. The role label is read from the corrupted input, never from the clean target. A selected router coordinate is replaced at the specified tokens and interfaces; the branch, subsequent states, and unselected coordinates are then recomputed normally. In a role swap, masked positions receive the visible calibration mean and vice versa. The instrumented intact path was checked against the original logits in FP32 and BF16.

For sequence s, let $L_s(c)$ be its summed target loss under condition c, and N_s its valid target count. Each of the 5,000 paired sequence bootstrap resamples B computes

$$\widehat{\mathrm{CE}}_B(c) = \frac{\sum_{s \in B} L_s(c)}{\sum_{s \in B} N_s}, \qquad \Delta_B(c) = \widehat{\mathrm{CE}}_B(c) - \widehat{\mathrm{CE}}_B(\mathrm{intact}). \tag{3}$$

Intervals and gain ratios are calculated within the same paired resample.

For two-stream write interventions, let $V = g_0 + g_1$, $L(V) = \max(0, V-2)$, and $U(V) = \min(2, V)$. We replace total gain V and bounded destination $q_i^{\mathrm{box}} = (g_i - L(V))/(U(V) - L(V))$ separately, then recover $g_i' = L(V') + [U(V') - L(V')]q_i^{\mathrm{box}\prime}$. This keeps both gains in the full router's valid range without clipping. When $V \le 2$, $q_i^{\mathrm{box}} = g_i/V$; when $V > 2$, the coordinates differ. The latter regime covers 14.8% of masked calibration token–interface pairs at $r = 0.5$.

B STREAM WIDTH AND TRANSFER

Table 2: Final in-domain validation at update 53,399. Learned $n = 1$ retains token-adaptive scalar gates. Lower CE is better.

Interface	Parameters	MDLM CE	AR CE
Standard residual	336.1M	3.605514	3.018551
Learned $n = 1$	336.2M	3.580209	3.014518
$n = 2$	336.6M	3.546667	2.986202
$n = 3$	337.4M	**3.541492**	2.966527
$n = 4$	340.4M	3.542203	**2.957888**

The $n = 2$ MDLM gain is 0.058847 CE from the standard residual, or 91.9% of the largest observed reduction (0.064022 at $n = 3$). The difference between $n = 2$ and the best wider result is only 0.005175 CE, establishing a clear saturation profile at this scale.

Table 3: External masked-reconstruction CE change from the standard residual, with paired 95% sequence-bootstrap intervals. All $n \geq 2$ models improve on every corpus; $n = 2$ has the largest mean percentage reduction across corpora.

Corpus	Streams	ΔCE	95% interval
OpenWebText	$n = 1$	$-.0145$	$[-.0190, -.0100]$
	$n = 2$	$-.0472$	$[-.0521, -.0423]$
	$n = 3$	$-.0566$	$[-.0618, -.0516]$
	$n = 4$	$-.0528$	$[-.0583, -.0476]$
WikiText-103	$n = 1$	$-.0118$	$[-.0178, -.0058]$
	$n = 2$	$-.0791$	$[-.0848, -.0735]$
	$n = 3$	$-.0635$	$[-.0702, -.0564]$
	$n = 4$	$-.0336$	$[-.0395, -.0275]$
LAMBADA	$n = 1$	$+.0015$	$[-.0026, +.0057]$
	$n = 2$	$-.0306$	$[-.0350, -.0263]$
	$n = 3$	$-.0230$	$[-.0275, -.0184]$
	$n = 4$	$-.0271$	$[-.0313, -.0229]$
C4	$n = 1$	$-.0178$	$[-.0224, -.0133]$
	$n = 2$	$-.0544$	$[-.0600, -.0488]$
	$n = 3$	$-.0531$	$[-.0584, -.0480]$
	$n = 4$	$-.0554$	$[-.0611, -.0499]$

The $n = 2$ improvement is positive on all four external corpora. Its relative exponential-loss reductions are 4.61% on OpenWebText, 7.61% on WikiText-103, 3.02% on LAMBADA, and 5.29% on C4, for a 5.13% unweighted mean. Larger width improves some individual corpora, while $n = 2$ gives the strongest mean transfer in this sweep.

C ROLE-FACTORIZED ROUTING LEARNED FROM SCRATCH

FRR trains two persistent streams with a compact interface. Each attention and MLP site learns a masked/visible-specific read vector and softmax write destination d_i. A scalar predictor of the current two-stream state sets *token-dependent* volume v, with physical write gains $g_i = 2vd_i$. The old-state mixer is the identity. FRR tests whether role-conditioned routing together with dynamic volume captures the principal benefit of a fully dynamic two-stream interface. This is a separate training run, not an intervention on the full checkpoint. It uses the same training seed, data order, corruption masks, backbone, schedule, and validation blocks as the width comparison.

Table 4: Final MDLM validation CE by corruption ratio. The mean weights the five ratios equally.

Interface	$r = .1$	$r = .3$	$r = .5$	$r = .7$	$r = .9$	Mean
Residual	1.7060	2.2986	3.2581	4.5212	6.2436	3.605514
Learned $n = 1$	1.6760	2.2593	3.2229	4.4990	6.2439	3.580209
FRR	1.6518	2.2550	3.2054	4.4759	6.2288	3.563375
Full $n = 2$	1.6212	2.2361	3.1823	4.4649	6.2289	3.546667

FRR gains 0.042139 CE over the standard residual, reproducing 71.6% of the full $n = 2$ gain. It also improves on learned $n = 1$ by 0.016834 CE, closing 50.2% of the gap from $n = 1$ to full $n = 2$. Its advantage over the standard residual is present at every recorded validation milestone from 5,000 updates onward. The trained comparison supports the joint usefulness of a second stream, role-based read and write destination, and a state-dependent scalar volume.

D Role routing in the full two-stream checkpoint

At $r = 0.5$, intact $n = 2$ has CE 3.20422 and the separately trained standard residual has CE 3.26993 on the same 1,024 test sequences. In Table 5, each row changes one control throughout all interfaces and positions while the others stay dynamic. The joint row changes the four controls together. Role rescue is the fraction $(\mathrm{CE}_{\mathrm{global}} - \mathrm{CE}_{\mathrm{role}})/(\mathrm{CE}_{\mathrm{global}} - \mathrm{CE}_{\mathrm{intact}})$.

Table 5: Increase in masked-target CE over intact $n = 2$, with paired 95% intervals. Global means discard role; role means retain masked/visible assignment; swaps exchange it.

Control	Setting	ΔCE	95% interval
Read	Global	$+.9493$	$[+.9406, +.9575]$
	Role	$+.0405$	$[+.0392, +.0419]$
	Swap	$+1.8108$	$[+1.7970, +1.8241]$
Write destination	Global	$+1.2551$	$[+1.2391, +1.2712]$
	Role	$+.0171$	$[+.0162, +.0181]$
	Swap	$+2.9771$	$[+2.9471, +3.0062]$
Write volume	Global	$+2.0262$	$[+2.0044, +2.0471]$
	Role	$+.2003$	$[+.1959, +.2046]$
	Swap	$+3.1725$	$[+3.1431, +3.2008]$
Transport	Global	$+.0194$	$[+.0183, +.0204]$
	Role	$+.0058$	$[+.0053, +.0063]$
	Swap	$+.0431$	$[+.0415, +.0446]$
All jointly	Global	$+4.5115$	$[+4.4759, +4.5465]$
	Role	$+1.6453$	$[+1.6242, +1.6659]$
	Swap	$+4.7515$	$[+4.7180, +4.7848]$

Read and destination show the clearest role structure: role means recover 95.7% and 98.6% of the corresponding global-mean damage, respectively. Swapping the assignments makes both substantially more damaging. Write volume additionally carries a consequential $+0.2003$ CE within-role component. Joint replacement is a stronger intervention because all coordinates become fixed within a role and cannot adapt to states changed by the other replacements.

E Token-matched write volume and depth

Matching volume to tokens. A role-mean replacement removes within-role variation; a within-role shuffle preserves the measured distribution but moves values among tokens of the same role. Both interventions leave all other router controls dynamic. Table 6 shows that token assignment matters beyond the role-specific volume distribution.

Table 6: Volume intervention at $r = 0.5$: paired CE increase over the intact checkpoint, with 95% intervals.

Positions changed	Role mean	Within-role shuffle
Masked	$+.0684$ $[+.0665, +.0703]$	$+.1322$ $[+.1300, +.1345]$
Visible	$+.1242$ $[+.1206, +.1276]$	$+.2939$ $[+.2886, +.2992]$

Localization through depth. Replacing volume only in the first block raises CE by $+.0673$ $[+.0651, +.0694]$ at visible positions and $+.0015$ $[+.0013, +.0018]$ at masked positions. In the last block the pattern reverses: masked replacement raises CE by $+.0181$ $[+.0172, +.0190]$, while visible replacement has no measured effect. The complete per-block intervention is plotted in Figure 3b. This causal handoff aligns early visible-evidence processing with later masked-target refinement.

Table 7: Network-wide role-mean volume replacement at different mask ratios. Each entry is a paired CE increase over its intact checkpoint; 128 sequences are used per ratio.

Mask ratio	Masked positions	Visible positions	Both roles
0.1	$+.0481$	$+.1220$	$+.1769$
0.3	$+.0556$	$+.1376$	$+.2014$
0.5	$+.0621$	$+.1256$	$+.1972$
0.7	$+.0547$	$+.0882$	$+.1501$
0.9	$+.0302$	$+.0349$	$+.0672$

The full-network effect peaks under partial observation and declines toward $r = 0.9$. The both-roles intervention is measured directly; it is not obtained by adding the two single-role effects.

E.1 Autoregressive control

State-dependent volume also affects AR checkpoints. On 1,024 matched AR sequences, replacing volume by an interface mean raises CE by $+.5837$ $[+.5756, +.5918]$ for $n = 2$, $+.3648$ $[+.3552, +.3745]$ for $n = 3$, and $+.3831$ $[+.3745, +.3918]$ for $n = 4$. Even exact-position means leave effects of $+.5466$, $+.3574$, and $+.3631$, respectively. This control identifies dynamic volume as a general residual mechanism; the distinctive diffusion structure is the strong causal assignment of read and write routes by observed corruption role.

F First-block construction and evidence transfer

Addressing the first update. The first attention interface, a_0, sends most of its write to s_{E} at visible tokens (mean destination shares $(0.88, 0.12)$) and to s_{P} at masked tokens ($(0.02, 0.98)$). In the masked-role edge intervention, deleting the a_0 write raises CE from 3.2043 to 8.0761. Restoring only its contribution to the following MLP read gives CE 3.2900, recovering 98.24% of the deletion effect. A separate destination test keeps the branch output and total gain fixed while reassigning the two write gains. Sending the visible a_0 write to the opposite stream costs $+3.0416$ CE; sending it to the learned stream costs only $+.0271$. The corresponding masked-role effects are $+.4623$ and $+.0017$. These tests establish that the early state depends causally on where the update is placed.

Source-to-target patching. For each masked target, we mask candidate visible sources and retain pairs whose source ablation changes the clean target log probability by at least 0.5 nats. The candidate pool contains six nearest visible positions and two random positions; the source is the candidate with the largest effect. This gives 179 pairs across 91 sequences. After each interface we patch the clean state into either the visible source or the masked target and rerun the remaining network. Recovery is the fraction of the source-ablation effect removed by the patch. At a_1, restoring s_{E} at the masked target recovers 0.059 of the effect; by a_8 it recovers 0.889. Over the same interval, recovery from restoring both streams at the visible source falls from 0.962 to 0.516. Patching s_{P} at the target stays near 0.4–0.5 until the later reopening phase, while patching both target streams at a_8 recovers 0.983.

These paired counterfactuals locate visible-source influence in the target's evolving stream during the preservation interval.

G PRESERVATION AND LATE REUSE OF THE SECOND STREAM

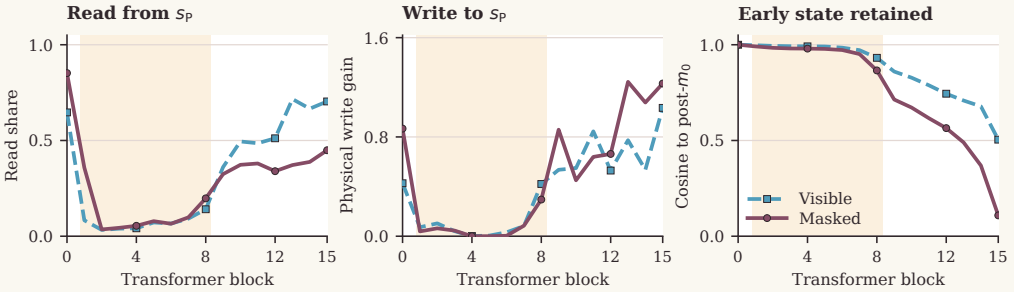

Figure 6: **A first-block state persists through the middle network.** The preserved stream receives little read or write mass through middle depth, while its cosine similarity to its own post-m_0 state remains high at masked and visible positions. Later read and write activity coincides with transformation of this sample-specific state.

State similarity and causal replacement tell the same story. At a_8, s_P has cosine similarity 0.949 to its own post-m_0 masked state and 0.965 at visible positions. Replacing $s_P^{a_8}$ with the *same example's* $s_P^{m_0}$ raises CE by only $+.0037$ [$+.0029, +.0045$] for masked positions and $+.0059$ [$+.0048, +.0070$] for visible positions. Replacing it with the post-a_0 state is much more disruptive, particularly at visible positions ($+.5870$ CE), identifying the post-m_0 state as the preserved object.

Position identity remains essential. Shuffling s_P among same-role tokens within each sequence after a_8 raises CE by $+.0820$ [$+.0777, +.0864$] for masked positions and $+.3721$ [$+.3495, +.3962$] for visible positions. Restoring the matched post-m_0 state removes 95% and 98% of this shuffle damage, respectively. Thus, the state remains position-specific within the sequence.

Read and write mass on s_P increase again after roughly m_8–m_{10}. Its masked-position cosine similarity to the post-m_0 state falls from 0.949 at a_8 to 0.672 at m_{10} and 0.109 at m_{15}, consistent with transformation during reuse. Late role-specific interventions in the main text show that visible tokens can rely primarily on the retained state, while masked predictions use both streams.

Table 8: Late stream-content intervention after m_{12}. Both streams at the selected role are set to either s_E or s_P before the remaining network is rerun. Values are paired CE increases over intact $n = 2$, with 95% intervals.

Role	Both set to s_E	Both set to s_P
Visible	$+.1475$ [$+.1356, +.1604$]	$+.0104$ [$+.0087, +.0121$]
Masked	$+.0772$ [$+.0741, +.0804$]	$+.2304$ [$+.2184, +.2432$]

At visible positions the retained state is the stronger late carrier, while masked predictions require distinct content from both streams. Removing s_P immediately before the output head adds $+.3549$ CE [$+.3402, +.3704$] on masked targets, confirming its contribution at readout.

H PRESERVATION ACROSS WIDTHS AND THE AR COMPARISON

We repeat the routing survey on 64 common sequences for diffusion $n = 2, 3, 4$ and their AR counterparts. For each model and role, Table 9 identifies the stream with the smallest mean read share in the a_2–a_8 window, then measures its later read share over a_{11}–m_{15}. The stream is selected by the middle-window rule before comparing the late values.

Table 9: Read share of the most preserved candidate stream. Diffusion candidates have particularly low middle-depth read share and are read more heavily near output.

Objective	Streams and role	Middle read share	Late read share
MDLM	$n = 2$, masked / visible	.062 / .059	.386 / .617
MDLM	$n = 3$, masked / visible	.062 / .086	.598 / .639
MDLM	$n = 4$, masked / visible	.051 / .075	.301 / .368
AR	$n = 2$	.473	.207
AR	$n = 3$	.167	.632
AR	$n = 4$	.117	.560

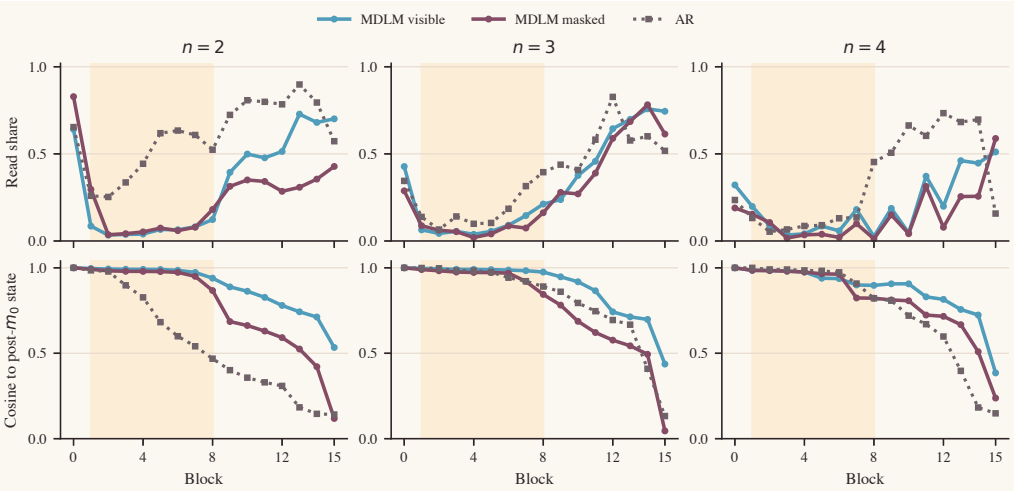

Figure 7: **The full preservation trajectory is characteristic of diffusion.** For each width, the top row gives the candidate stream's read share and the bottom row its cosine similarity to the post-m_0 state. Diffusion is separated by visible and masked roles; AR has a single position regime. Shading marks the middle-depth preservation interval.

The diffusion pattern recurs at all three widths and both token roles: middle-depth read share stays below 0.10, then rises sharply late. The AR $n = 2$ candidate is read throughout the middle network and its early-state similarity falls rapidly. AR $n = 3, 4$ can show late increases in read share, but their candidate streams are read more during the middle interval and depart sooner from the post-m_0 state (Figure 7). The combined preservation–transfer–reuse trajectory is therefore nearly absent in AR and consistently present in all tested diffusion models.