Mechanism Shift During Post-training from Autoregressive to Masked Diffusion Language Models

Anonymous ACL submission

Abstract

Post-training pretrained autoregressive models (ARMs) into masked diffusion models (MDMs) has emerged as a cost-effective way to overcome the limitations of sequential generation. Yet it remains unclear whether post-trained MDMs acquire genuinely new computational mechanisms or merely re-express autoregressive computation in a non-autoregressive form. Through a comparative circuit analysis of ARMs and their MDM counterparts post-trained from the same backbones, we uncover two complementary axes of reorganization. Structurally, the shift is task-dependent: MDMs preserve autoregressive circuitry on locally causal tasks but abandon inherited pathways and front-load computation into early layers on global tasks. Semantically, the shift is consistent across regimes: sharp, localized specialization in ARMs gives way to distributed integration in MDMs. Together, these findings show that diffusion post-training is not a surface-level change in the generation procedure but a reorganization of internal computation whose depth depends on the task.

1 Introduction

Large language models have achieved near-human performance across diverse linguistic tasks (Touvron et al., 2023; Qwen et al., 2025). Yet the autoregressive framework imposes structural limitations (Welleck et al., 2019): causal masking cannot revise earlier tokens, so early errors propagate throughout the sequence (Gu et al., 2018; Bengio et al., 2015; Ranzato et al., 2016). Moreover, many reasoning and planning tasks require global reasoning, where early decisions must satisfy constraints on the entire sequence (Ye et al., 2025a).

Masked diffusion models (MDMs) have gained interest as a non-autoregressive paradigm well-suited to overcoming these limitations (Austin et al., 2021; Sahoo et al., 2024), but training them from scratch remains computationally expensive due to slower convergence (Gong et al., 2025). To mitigate this cost, recent work post-trains pretrained autoregressive models (ARMs) into the diffusion paradigm (Gong et al., 2025; Ye et al., 2025b), with models such as Dream (Ye et al., 2025b) achieving strong performance at a fraction of the compute needed for training from scratch.

Despite the empirical success of post-training ARMs with diffusion objectives (Gong et al., 2025), the resulting algorithmic changes remain poorly understood. Since adopting a new training objective does not by itself guarantee a corresponding internal mechanism (Prakash et al., 2024), a central mechanistic question arises: does diffusion post-training uniformly preserve inherited autoregressive pathways, or does it selectively rewire them depending on the structural demands of the task?

In this work, we address this question through circuit-level analysis (Bhaskar et al., 2024) of ARMs and MDMs post-trained from the same autoregressive backbones. We first investigate **where** algorithmic changes occur by comparing task-specific circuit structures across paired ARM–MDM models and then examine **how** these changes are realized through component-wise logit-lens probes and neuron-level activation analyses.

Our analysis uncovers a "mechanism shift" along two complementary axes. At the circuit level, the shift is task-dependent: MDMs preserve autoregressive circuitry on locally causal tasks but abandon the inherited pathways and front-load computation into early layers when global planning is required. At the semantic level, the shift is consistent across regimes: sharp, localized specialization in ARMs gives way to distributed integration in MDMs. Together, these results show that diffusion post-training is not a surface-level change in the generation procedure but a genuine reorganization of internal computation.

2 Related Works

2.1 Masked Diffusion Models

MDMs generate text by reversing a corruption process that stochastically replaces tokens with a [MASK] symbol (Chang et al., 2022; Austin et al., 2021), enabling non-autoregressive generation with full bidirectional context. While training such models from scratch (Nie et al., 2025) is computationally expensive, recent work has shown that pretrained ARMs can be effectively post-trained into MDMs (Gong et al., 2025). Rather than learning diffusion dynamics from scratch, these approaches initialize from a pretrained ARM and post-train it to iteratively denoise partially masked inputs.

From DiffuLLaMA (Gong et al., 2025) to Dream (Ye et al., 2025b), this line of work shows that such post-training retains the practical advantages of diffusion—parallel decoding, iterative refinement, and bidirectional attention—while reducing training cost, and achieves strong performance on directionality-sensitive tasks. However, these works focus on performance and efficiency, leaving open the question of how diffusion objectives reshape the underlying computational mechanisms.

2.2 Mechanistic Interpretability and Circuits

Mechanistic interpretability aims to identify the internal components and algorithms responsible for specific model behaviors (Olah et al., 2020; Elhage et al., 2021). A central concept is the **circuit**, defined as a subgraph of the computational graph connecting inputs to the unembedding projection that is sufficient to produce a target behavior (Olah et al., 2020; Bhaskar et al., 2024). Nodes correspond to components such as attention heads and MLPs, while directed edges represent causal dependencies between components, where the output of one node contributes to the input of another (Ou et al., 2025; Hanna et al., 2024).

Empirical studies show that behaviors can be explained by sparse circuits involving a small fraction of model connections (Bhaskar et al., 2024; Wang et al., 2023). Methods such as Edge Attribution Patching identify these subgraphs via gradient-based attribution, enabling circuit discovery for tasks including indirect object identification and numerical comparison (Hanna et al., 2023; Lieberum et al., 2023; Bhaskar et al., 2024). Across models and scales, similar circuits, such as induction heads recur consistently, suggesting that they implement stable algorithmic functions rather than incidental patterns (Prakash et al., 2024; Tigges et al., 2024; Ou et al., 2025; Wang et al., 2025). Recent automated approaches, including ACDC and EAP, enable scalable circuit discovery without manual inspection (Syed et al., 2024; Bhaskar et al., 2024).

While mechanistic analyses have begun to probe diffusion models in the vision domain (Shi et al., 2025; Niedoba et al., 2025), comparable studies for text diffusion models remain limited. As a result, it is unclear whether diffusion objectives induce distinct computational strategies in language models or reorganize existing autoregressive circuitry. We address this question by treating paired ARMs and post-trained MDMs as a controlled comparison, isolating mechanistic changes induced by diffusion post-training from those caused by architectural or initialization differences.

3 Method

Figure 1 illustrates our comparative framework for investigating the mechanistic shift from ARMs to MDMs. By comparing pretrained ARMs with MDMs post-trained from the same autoregressive backbones, we reduce confounds from model scale, initialization, and base architecture. This setup allows us to examine how the ARM-to-MDM transition reshapes internal computation, although the transition includes changes in both training objective and denoising interface.

3.1 Models and Configuration

To ensure the generalizability of our findings, we evaluate two model families: Qwen2.5-7B (Qwen et al., 2025) paired with Dream-Base-7B (Ye et al., 2025b), and LLaMA-2-7B (Touvron et al., 2023) paired with DiffuLLaMA-7B (Gong et al., 2025). Since each MDM is post-trained from its autoregressive backbone, we can directly attribute computational differences to diffusion post-training rather than to architectural variation.

3.2 Tasks and Datasets

To examine whether circuit reorganization depends on task structure, we distinguish between causal and global regimes. In causal tasks, the target can largely be inferred from left-to-right prefix context, whereas in global tasks it depends on constraints across the full sequence. Rather than relying on task-name intuition, we use a Dream-based entropy gap, $\Delta H := H_{\text{Dream}}(x_t \mid x_{<t}) - H_{\text{Dream}}(x_t \mid x_{\setminus t})$, where $x_{\setminus t}$ denotes the sequence with x_t

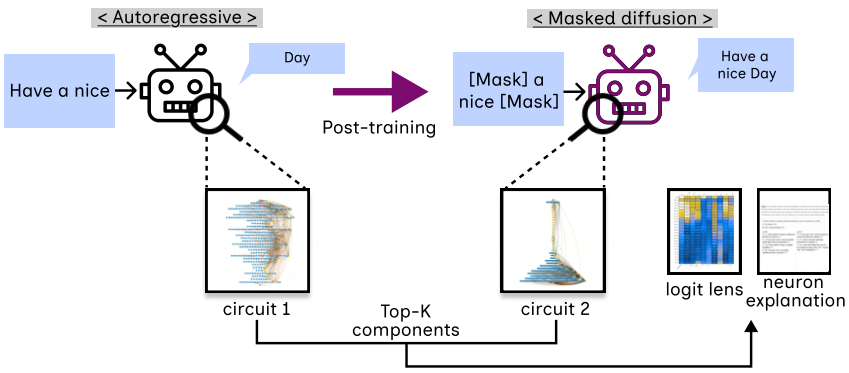


Figure 1: **Overview of the mechanism shift analysis pipeline.** We extract task-specific circuits for both the ARM baseline and the post-trained MDM. We compare their circuits at the edge, component, and layer levels, then analyze high-attribution *Top-K components* using *Logit Lens* and *Neuron Explanation*.

masked. Positive ΔH indicates a global regime, while non-positive ΔH indicates a causal or prefix-dominant regime. Using this criterion, we select two causal and two global tasks; details are provided in Appendix A.

3.2.1 Causal Tasks

Indirect Object Identification (IOI): IOI is a canonical circuit-analysis task in which the model identifies the correct indirect object in a short narrative with repeated names (Wang et al., 2023). Since the answer can be resolved from preceding context, IOI tests whether inherited autoregressive pathways are preserved under diffusion post-training.

Greater-Than (GT): GT is a numerical comparison task in which the model predicts a year greater than a reference year given in the prompt (Hanna et al., 2023). The constraint can be resolved directly from the prefix, making GT a complementary test of prefix-bound numerical reasoning.

3.2.2 Global Tasks

Countdown: Countdown requires constructing a valid arithmetic expression from a set of source integers to match a target value (Ye et al., 2025a). Since each intermediate choice must satisfy a final global constraint, Countdown tests the model's ability to plan beyond left-to-right generation.

Semantic Infilling (SI): SI requires predicting missing intermediate phrases conditioned on both preceding and subsequent text blocks. Because the missing span must be coherent with context on both sides, SI probes bidirectional coordination and non-sequential integration.

Generation budgets are aligned to task demands: causal tasks use one diffusion step, whereas global tasks use steps matched to the target sequence length. Details are provided in Appendix A.

3.3 Circuit Discovery and Analysis Pipeline

To trace where and how computation changes, we analyze circuits in three stages: discovery, attribution-guided comparison, and mechanism interpretation. Discovery extracts task-specific circuits for each model. Attribution-guided comparison quantifies structural overlap and depth-wise localization between paired ARMs and MDMs. Mechanism interpretation probes how task-critical components align with output vocabulary and what internal features they encode.

3.3.1 Discovery

Before comparing ARMs and MDMs, we first identify the task-specific circuit used by each model. Since both are transformer-based, their computation shares a common graph of attention heads, MLPs, and directed edges. We use Edge Attribution Patching with Integrated Gradients (EAP-IG) (Hanna et al., 2024) to score every edge by its contribution to the target behavior, and retain the top 1,000 edges by attribution magnitude as the task-specific circuit $\mathcal{C} \subset \mathcal{G}$. This fixed budget keeps the circuit sparse while preserving a connected pathway from input-side components to the output logits, enabling direct comparison across ARMs and MDMs.

3.3.2 Attribution-Guided Circuit Comparison

After discovering task-specific circuits for each model, we compare how computation is reorga-

Level	Metric	Interpretation
Edge	$J(\mathcal{E}^{\text{top}}_{\text{ARM}}, \mathcal{E}^{\text{top}}_{\text{MDM}})$	Pathway reuse
Component	$s(v) = \sum_{(u,v)\in\mathcal{E}^{\text{top}}} a_{u\to v} + \sum_{(v,u)\in\mathcal{E}^{\text{top}}} a_{v\to u}$	Influential source/sink components
Layer	$\text{CoG} = \frac{\sum_l lA_l}{\sum_l A_l}$	Depth-wise localization

Table 1: Summary metrics for attribution-guided circuit comparison. Here, $\mathcal{E}^{\text{top}}$ denotes the retained top-attribution edges, $a_{u\to v}$ denotes the EAP-IG attribution score of edge $u \to v$, and $A_l = \sum_{v:\ell(v)=l} s(v)$ denotes the total attribution mass assigned to layer l.

nized when an ARM is post-trained into an MDM. Our goal is to measure whether the MDM reuses the same high-attribution pathways as the ARM, shifts responsibility to different components, or relocates computation to different layers. We therefore compare each discovered circuit at three complementary levels: edges, components, and layers, as shown in Table 1.

Edge-level overlap. For each ARM–MDM pair, we select the highest-attribution edges under EAP-IG (Hanna et al., 2024) on identical prompts and compute their Jaccard overlap. High overlap indicates pathway reuse, whereas low overlap suggests that diffusion post-training recruits different computational routes. We use the top 1000 edges in all experiments, balancing attribution coverage and circuit sparsity; details are provided in Appendix C.

Component-level importance. Edge overlap tells us whether two models use similar connections, but not which components dominate the computation. We therefore aggregate EAP-IG attribution scores over the incoming and outgoing high-attribution edges incident to each node v, yielding a component score $s(v)$. Components with large $s(v)$ repeatedly appear as sources or sinks of important information flow. We define the Top-K components as the $K = 100$ nodes with the largest scores and use them as the focus of downstream logit-lens and neuron-level analyses. The choice of K is discussed in Appendix C.

Layer-wise localization. Finally, to summarize where the circuit is concentrated across depth, we compute a layer-wise Center of Gravity (CoG). Lower CoG indicates that the attribution mass is concentrated in earlier layers, while higher CoG indicates greater reliance on middle or late layers. This provides a scalar summary of whether diffusion post-training preserves the depth profile of the ARM circuit or shifts computation toward earlier layers. Formal details are provided in Appendix D.

3.3.3 Mechanism Interpretation

After identifying where circuits differ, we next ask how the computation carried by these changed components differs. We focus on the Top-K EAP-IG components from the attribution-guided comparison and analyze them using two complementary probes: component-wise logit lens and neuron-level explanation.

Component-wise Logit Lens. Following the logit lens framework (nostalgebraist, 2020), we project intermediate component activations into vocabulary space using each model's native unembedding matrix. Because each MDM is post-trained from the same autoregressive backbone as its paired ARM and retains the same tokenizer, the paired models share an identical vocabulary space. We compute diagnostic logits as $z^{(c,t)}_{\text{AR}} = (W^{\text{AR}}_U)^\top h^{(c,t)}_{\text{component}}$ for ARMs and $z^{(c,t,s)}_{\text{MDM}} = (W^{\text{MDM}}_U)^\top h^{(c,t,s)}_{\text{component}}$ for MDMs, where c indexes the component, t the token position, and s the diffusion timestep. We trace ARM alignments across token positions and MDM alignments across diffusion timesteps at fixed positions.

These projections are used only as diagnostic output-basis probes, not as calibrated predictions or basis-invariant evidence. Thus, they indicate which vocabulary directions a component aligns with, rather than proving that the component directly predicts those tokens.

To quantify these patterns, we compute summary statistics from component-wise logit distributions. For IOI, we measure name-token concentration with *NameFrac@K*. Across tasks, we measure specialization using *LogitGap*, ΔLME, and top-K entropy. Details are provided in Appendix D.

Neuron Explanation. To complement output-space probes, we examine the internal features encoded by high-attribution components, focusing on the early layers where MDM circuits place substantial attribution mass. We define a neuron as

a single scalar coordinate in the residual stream at a given transformer layer. For each task, we record activations over neuron indices and token positions in layers 0–4, restricted to components in the Top-K EAP-IG circuit, and following Bills et al. (2023) collect the maximally activating input contexts for qualitative and automated explanation. For MLPs we inspect top-activating residual coordinates; for attention components we analyze head output features. Because individual coordinates are not basis-invariant, we interpret neuron explanations only at the aggregate category level rather than as definitive labels for single neurons.

Together, component-wise logit lens and neuron-level explanation provide complementary views of the mechanism shift: the former characterizes how influential components align with output vocabulary directions, while the latter examines what internal features support those alignments.

4 Results & Analysis

4.1 Circuit-Level Differences

Figure 2 shows circuits for ARMs and MDMs on IOI and COUNTDOWN. The corresponding circuit visualizations for GREATER-THAN and SEMANTIC INFILLING are provided in Appendix C. Combined with the overlap and CoG metrics in Table 2, including the GREATER-THAN(GT) and SEMANTIC INFILLING(SI) evaluations, the results reveal a task-dependent pattern of circuit reuse and reorganization.

Task	Model Pair	Overlap		CoG (Layer)	
		Edge	Top-K	ARM	MDM
Causal regime					
IOI	Qwen / Dream	0.193	0.105	17.5	20.4
	LLaMA / DiffuLLaMA	0.088	0.124	18.2	19.8
GT	Qwen / Dream	0.290	0.515	18.7	14.4
	LLaMA / DiffuLLaMA	0.054	0.117	17.3	11.7
Global regime					
COUNTDOWN	Qwen / Dream	0.008	0.093	16.5	4.8
	LLaMA / DiffuLLaMA	0.032	0.081	17.1	5.3
SI	Qwen / Dream	0.225	0.429	13.7	3.8
	LLaMA / DiffuLLaMA	0.018	0.124	13.1	3.9

Table 2: Circuit similarity and Center of Gravity (CoG) between ARMs and their MDM counterparts, grouped by task regime. Causal tasks show greater pathway retention or milder depth shifts, whereas global tasks show stronger early-layer relocation, with the degree of overlap varying across model families. Higher overlap indicates reuse of autoregressive circuitry.

For IOI, MDMs preserve the autoregressive circuitry inherited from their ARM initializations. Table 2 quantifies this structural retention using edge overlap and CoG. Although the overlap score appears small (e.g., 0.193 for Qwen/Dream), causal ablation shows that the shared edges form the functional core of the circuit: ablating the intersection reduces IOI accuracy to 28.1%, whereas ablating a random subset of non-shared edges of equal size retains $92.4 \pm 1.2\%$ accuracy across five random seeds. The CoG remains relatively stable in the mid-to-late layers (17.5 vs. 20.4), indicating that causal reasoning machinery learned during pre-training is reused rather than replaced.

In contrast, for COUNTDOWN, the inherited autoregressive circuitry is much less reused. Circuit overlap drops sharply (e.g., to 0.008 for Qwen/Dream), indicating structural decoupling. This change is accompanied by a strong shift in depth, with the CoG moving from layer 16.5 in the ARM to 4.8 in the MDM. This depth shift is also consistent with the aggregated layer-wise component usage statistics, which show increased concentration in earlier layers (Appendix C Figure 3).

The complementary GT and SI results corroborate this regime-level dichotomy while revealing a secondary axis of variation across model families. GT, like IOI, behaves as a causal task: its circuit remains closer to the ARM regime than the global tasks, either through higher overlap, milder depth shifts, or both. SI, like COUNTDOWN, behaves as a global task: even when component overlap remains nontrivial, computation shifts strongly toward earlier layers, consistent with the demands of bidirectional integration. Within this regime-level pattern, the magnitude of reorganization varies across model families. Dream tends to preserve more of its Qwen backbone on causal tasks, whereas DiffuLLaMA shows greater structural decoupling from its LLaMA backbone across regimes. Nevertheless, both model families preserve the same qualitative split: causal tasks retain more inherited structure, while global tasks induce stronger early-layer relocation. This suggests that the depth of mechanism shift reflects both task structure and the post-training trajectory of each MDM.

Taken together, these findings show that masked diffusion post-training does not simply overwrite autoregressive mechanisms. Instead, it preserves inherited circuitry where it remains task-compatible, while recruiting early-layer components when global or bidirectional reasoning requires stronger non-sequential integration.

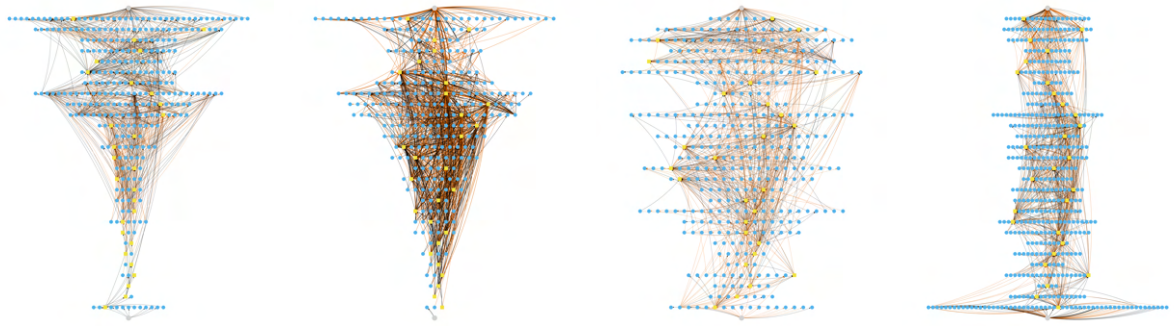


Figure 2: Circuit comparison across tasks and architectures. From left to right: IOI (Qwen2.5-7B), IOI (Dream-Base-7B), COUNTDOWN (Qwen2.5-7B), and COUNTDOWN (Dream-Base-7B). Node labels have been omitted to emphasize the global topological patterns rather than individual components. Blue nodes represent attention components (Q, K, V), while yellow nodes represent MLPs. Information flows from the input (bottom) to the final logits (top). Higher resolution visualizations are provided in the Appendix Figure 4.

Category	Dream	DiffuLLaMA	Qwen	LLaMA
PERSON	1,300	337	298	528
OTHER	8,348	3,251	2,965	14,606
ORG	62	30	81	418
LOC	55	33	31	343
DIGIT	32	–	2	55
OPERATOR	20	2	210	124
MISC	3	2	2	–
Total	**9,820**	**3,933**	**3,775**	**15,619**
NameFrac@10	**13.2%**	**8.6%**	**7.9%**	**3.4%**

Table 3: NER category distribution among the top-10 aligned tokens (T_{10}) for each model. NameFrac@10 represents the percentage of PERSON tokens relative to the total top-10 tokens extracted.

4.2 Semantic Reorganization of Task-Critical Components and Early Layers

To understand how circuit-level differences translate into concrete computational strategies, we analyze model behavior at two complementary granularities. First, we examine *task-critical components*—attention heads and MLPs that receive high EAP-IG attribution—using component-wise logit lens analysis. This allows us to characterize how semantic roles are assigned to components that directly influence task outputs. Second, we analyze *early-layer neuron activations* to understand how these semantic roles are implemented internally in regions where MDMs emphasize computation but component-level semantics appear diffuse.

4.2.1 Task-Critical Components: Component-wise Logit Lens Analysis

We analyze task-critical components identified by high EAP-IG attribution using a component-wise logit lens. This allows us to test whether task-relevant information is concentrated in a few specialized components or distributed across many. We report probes of semantic alignment and dominance in the main text, and defer representative top-token examples to Tables 4 and 9.

Causal reasoning task (IOI). For IOI, ARMs exhibit *sharply localized semantic specialization*: a small set of high-attribution components aligns decisively with person-name tokens. As shown in Table 3, MDMs can place person-name tokens relatively often among their top aligned candidates—Dream reaches a higher NAMEFRAC@10 (13.2%) than Qwen (7.9%) and LLaMA (3.4%), with DiffuLLaMA at 8.6%—but this does not imply sharper specialization. ARMs exhibit substantially stronger selective amplification once a name token appears, with median $\Delta\mathrm{LME}$ values of 0.90 for Qwen and 0.89 for LLaMA, compared with 0.31 for Dream and 0.05 for DiffuLLaMA (Appendix Table 12). ARMs also show larger logit gaps, while DiffuLLaMA exhibits higher entropy. Thus, raw name frequency does not imply specialization: ARMs remain dominated by a few decisive name-specific components, whereas MDMs spread name-related evidence across broader component sets.

Representative components mirror this pattern (Table 4). High-attribution attention heads in Qwen and LLaMA align strongly with person-name tokens, consistent with pointer-like behavior, and DiffuLLaMA largely preserves this pattern. Dream, however, departs from it: its high-attribution components are less clearly name-specific, often aligning with non-person proper nouns or broader se-

Task	Model	Component	Mean Logit	Top Tokens	Role
IOI	Qwen	a27.h5	21.14	Dan	Person-name-related Component
	Qwen	a23.h11	6.84	Ben	Person-name-related Component
	LLaMA	a24.h15	2.75	Jerry	Person-name-related Component
	LLaMA	a21.h1	1.73	Carol	Person-name-related Component
	Dream	m25	3.14	Browser	Proper-noun component
	Dream	m22	2.33	Kremlin	Proper-noun component
	DiffuLLaMA	a26.h21	2.81	Marian	Person-name-related Component
	DiffuLLaMA	a22.h19	1.80	Grace	Person-name-related Component
COUNTDOWN	Qwen	m25	51.29	3	Digit-related Component
	Qwen	m20	20.67	1, 2	Broad Numerical Component
	LLaMA	m29	6.20	pick	Instruction-related Component
	LLaMA	a22.h13	2.10	four	Numerical–Lexical Component
	Dream	m27	26.06	1, 2, 3	Broad Numerical Component
	Dream	m23	5.34	5, 1, 4	Broad Numerical Component
	DiffuLLaMA	m31	12.32	0, 1, 2	Broad Numerical Component
	DiffuLLaMA	m4	1.40	–	Symbol-related Component

Table 4: Representative components exhibiting high logit concentration across tasks and model families. Roles are descriptive labels summarizing observed logit distribution patterns rather than definitive functional assignments.

mantic tokens. Even when circuit topology is similar, diffusion post-training can redistribute semantic roles and weaken the dominance of individual name-specific heads.

Global reasoning task (COUNTDOWN). A contrasting pattern emerges in COUNTDOWN. Whereas ARMs rely on components with *strong numerical selectivity*—sharply aligned with specific digits or operators and suggesting that global planning is approximated through sequential, component-centric heuristics (Table 4)—MDMs dissolve this concentration. In Dream, the sharp specialization collapses entirely: instead of a single dominant numerical component, multiple components exhibit moderate responses across numerical tokens with no individual component exerting decisive control. DiffuLLaMA occupies an intermediate regime, retaining numerical associations but with reduced magnitude and increased dispersion. This redistribution is also architectural—both MDMs front-load task-relevant computation, with CoG dropping to 4.8 for Dream and 5.3 for DiffuLLaMA from mid-network ARM values (Table 2). To ensure that this early-layer localization is not merely a logit-lens artifact, we perform targeted causal ablations on the identified circuits. Isolating these early pathways triggers a catastrophic collapse in COUNTDOWN accuracy, establishing their strict causal necessity for global planning (see Appendix B for full metrics and control baselines).

Expanded GT/SI probes. The completed Top-100 component-filtered logit-lens metrics for GT and SI show the same broad pattern of reduced selective amplification in MDMs (Appendix Table 11). Qwen remains sharply concentrated on both tasks, with large LOGITGAP values on GT (3.107) and SI (3.050), whereas Dream's gaps are much smaller (0.085 and 0.042). LLaMA and DiffuLLaMA are both more diffuse than Qwen, but DiffuLLaMA remains at the low-gap, high-entropy end of the spectrum, especially on SI (LOGITGAP 0.076; entropy 2.263). Thus, the component-level reorganization observed on IOI and COUNTDOWN generalizes across both regimes: even when circuit overlap is nontrivial, diffusion post-training weakens single-component dominance and pushes task evidence toward broader component ensembles.

Taken together, these results indicate that diffusion post-training induces a *semantic reorganization at the component level*. Whereas ARMs resolve tasks through a small number of highly specialized components, MDMs redistribute semantic responsibility across multiple components, yielding a more ensemble-like computation—accompanied on global tasks by a pronounced shift of computation toward earlier layers.

4.2.2 Early-layer Representation: Neuron-level Analysis

While component-wise logit lens analysis captures semantic specialization among task-critical components, it does not fully explain the behavior of early-layer regions emphasized by MDM circuits. To characterize how task-relevant information is implemented there, we analyze neuron activations in the lowest transformer layers.

This analysis is especially important for COUNT-

DOWN, where the CoG shift in Table 2 indicates that post-training moves substantial computation into the bottom of the network. In ARMs, early-layer neuron activation is *highly task-dependent*. In IOI, many early-layer neurons are strongly activated by descriptive adjectives and modifier-related tokens (e.g., 2,821 neurons in LLaMA; Appendix Table 6), suggesting these layers encode syntactic features before resolving entity identity. In COUNTDOWN, this pattern shifts toward neurons associated with numerical or technical content (over 1,000 neurons). Despite this task-dependent reallocation, ARMs consistently exhibit strong concentration within specific semantic categories.

MDMs display a qualitatively different pattern. Across both tasks, the number of active neurons in early layers remains relatively stable (typically $\sim$50–200 active neurons per top category; Appendix Table 6), and these neurons tend to correspond to broad, genre-level cues rather than sharply defined task-specific categories. This suggests MDMs rely less on category-specific early-layer specialization and instead maintain a more uniform, task-agnostic representational regime. Combined with the CoG results, this pattern implies that MDM early layers absorb more computation without relying on sharply specialized neurons.

4.2.3 Connecting Component-level and Neuron-level Perspectives

Taken together, the component-level and neuron-level analyses reveal a mechanistic pattern. Component-wise logit lens characterizes how outputs are distributed across influential components, while neuron-level explanation provides aggregate evidence about early-layer representations. Because neuron labels from automated explanation are noisy, we interpret them only at the category-distribution level rather than as functional assignments for specific neurons.

In ARMs, automated explanations concentrate on a smaller set of interpretable units (266) but distribute very unevenly across semantic categories ($\sigma = 369.78$ in per-category counts): a few categories, such as descriptive adjectives or numerical values, account for a large fraction of explained early-layer neurons, while most categories attract only a handful (Appendix Table 6). This pattern suggests dominance by a limited set of highly specialized units. MDMs instead spread interpretable labels across more units (426) with substantially lower variance ($\sigma = 202.52$), reflecting more even coverage across semantic categories rather than concentration in a few. Both the larger unit count and the flatter category distribution point to broader, more diffuse responsibility.

This component-level reorganization is consistent with the circuit-level evidence: the front-loaded computation observed via CoG and the reduced dominance of interpretable components in logit-lens probes reflect complementary aspects of the same broader pattern. These findings suggest that diffusion post-training trades component-level specialization for distributed semantic coverage, a reorganization whose signature is visible at both the circuit and representational levels.

5 Conclusion

In this paper, we investigated where and how masked diffusion post-training reshapes the internal mechanisms of autoregressive language models. To identify *where* these changes occur, we used circuit analysis and layer-wise CoG to trace changes in circuit composition and depth-wise computation. To understand *how* they are realized, we used component-wise logit-lens probing and neuron-level analysis to examine shifts in semantic role assignment and representation patterns.

Our results reveal a mechanism shift along two complementary axes. Structurally, the shift is task-dependent: on causal tasks such as IOI and GREATER-THAN, MDMs largely preserve the inherited autoregressive circuitry, whereas on global tasks such as COUNTDOWN and SEMANTIC INFILLING, they abandon inherited pathways and front-load computation into early layers. Semantically, the shift is consistent across regimes: ARMs resolve tasks through a small number of sharply specialized components, while MDMs distribute semantic responsibility across broader component ensembles supported by early-layer representations with reduced task-specific selectivity.

Together, these findings show that masked diffusion post-training is not a surface-level change in the generation procedure but a genuine reorganization of internal computation—preserving autoregressive mechanisms for local causal reasoning while reorganizing them for more globally integrated, non-sequential reasoning. This regime-dependent reorganization suggests that training objectives reshape model internals only as deeply as the task demands, offering a useful lens for interpreting other post-training paradigms.

Limitations

Our evaluation covers four benchmarks (IOI, GREATER-THAN, COUNTDOWN, and SEMANTIC INFILLING) that contrast causal tracking with global planning. While this controlled setup isolates the mechanism shift along a clear regime axis, whether the same patterns generalize to more open-ended linguistic tasks remains to be tested. Second, we study the holistic adaptation regime in which the diffusion objective, iterative inference schedule, and post-training data exposure are coupled; disentangling the individual contribution of each factor is an important direction for future work. Finally, our attribution-guided pipeline isolates the most behaviorally influential pathways but does not exhaustively characterize all task-relevant computation; scaling the analysis to richer circuits and longer reasoning chains is left to future work.

References

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Adithya Bhaskar, Alexander Wettig, Dan Friedman, and Danqi Chen. 2024. Finding transformer circuits with edge pruning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. 2023. Language models can explain neurons in language models. https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. 2022. Maskgit: Masked generative image transformer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Scott Elshowk, Tristan Hume, Sam McCandlish, Pamela Mishkin, Danny Nguyen, Chris Olah, Eric Sigler, Kyle Sommer, and Ilya Sutskever. 2021. A mathematical framework for transformer circuits. https://transformer-circuits.pub/2021/framework/index.html. Transformer Circuits.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, Hao Peng, and Lingpeng Kong. 2025. Scaling diffusion language models via adaptation from autoregressive models. In *The Thirteenth International Conference on Learning Representations*.

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

Michael Hanna, Ollie Liu, and Alexandre Variengien. 2023. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. 2024. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. In *First Conference on Language Modeling*.

Tom Lieberum, Matthew Rahtz, J'anos Kram'ar, Geoffrey Irving, Rohin Shah, and Vladimir Mikulik. 2023. Does circuit analysis interpretability scale? evidence from multiple choice capabilities in chinchilla. *ArXiv*, abs/2307.09458.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025. Large language diffusion models. *Preprint*, arXiv:2502.09992.

Matthew Niedoba, Berend Zwartsenberg, Kevin Patrick Murphy, and Frank Wood. 2025. Towards a mechanistic explanation of diffusion model generalization. In *Forty-second International Conference on Machine Learning*.

nostalgebraist. 2020. Interpreting gpt: the logit lens. *lesswrong*.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*. Https://distill.pub/2020/circuits/zoom-in.

Yixin Ou, Yunzhi Yao, Ningyu Zhang, Hui Jin, Jiacheng Sun, Shumin Deng, Zhenguo Li, and Huajun Chen. 2025. How do llms acquire new knowledge? a knowledge circuits perspective on continual pre-training. In *ACL (Findings)*, pages 19889–19913.

Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. 2024. Fine-tuning enhances existing mechanisms: A case study on entity tracking. *ArXiv*, abs/2402.14811.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li,

Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks.

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

Yingdong Shi, Changming Li, Yifan Wang, Yongxiang Zhao, Anqi Pang, Sibei Yang, Jingyi Yu, and Kan Ren. 2025. Dissecting and Mitigating Diffusion Bias via Mechanistic Interpretability . In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8192–8202, Los Alamitos, CA, USA. IEEE Computer Society.

Aaquib Syed, Can Rager, and Arthur Conmy. 2024. Attribution patching outperforms automated circuit discovery. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 407–416, Miami, Florida, US. Association for Computational Linguistics.

Curt Tigges, Michael Hanna, Qinan Yu, and Stella Biderman. 2024. LLM circuit analyses are consistent across training and scale. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*.

Xu Wang, Yan Hu, Wenyu Du, Reynold Cheng, Benyou Wang, and Difan Zou. 2025. Towards understanding fine-tuning mechanisms of llms via circuit analysis. *ArXiv*, abs/2502.11812.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training.

Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong. 2025a. Beyond autoregression: Discrete diffusion for complex reasoning and planning. In *The Thirteenth International Conference on Learning Representations*.

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

A Task Details

Datasets For COUNTDOWN, we evaluate both LLaMA-series and Qwen-series models on 500 examples per model. LLaMA-series models use 13 diffusion steps (13 tokens), while Qwen-series models use 12 diffusion steps (12 tokens). For IOI, we evaluate 500 examples in a single-step setting with 1 diffusion step and 1 token. Since IOI in our setup requires generating only a single target token, increasing the number of diffusion steps would still correspond to repeatedly refining the same one-token prediction. As a result, the underlying circuit is not expected to change qualitatively across multiple steps, making the single-step setting the most direct and interpretable choice. For GREATER-THAN, we use the full 1,000-example evaluation set and align the diffusion budget to the two-token target continuation, averaging MDM circuits over 2 diffusion steps. For SEMANTIC INFILLING, we use the full 100-example evaluation set and align the diffusion budget to the tokenizer-specific target-span length: Qwen-series models use 3 diffusion steps, while LLaMA-series models use 4 diffusion steps.

Task Templates and Examples To ground our mechanistic evaluations, we detail the specific structural templates and input-output formats for each of the tasks below:

- **IOI (Indirect Object Identification):** This task tests local, directional linguistic structures by requiring the model to identify which name is the indirect object.
  - *Template:* "[Name A] and [Name B] went to the store. [Name A] gave a toy to [Target]"
  - *Example:* "John and Mary went to the store. John gave a toy to **Mary**"
- **COUNTDOWN:** This task evaluates global planning and arithmetic constraints. The model is given a target number and a set of allowed digits, and must generate a valid mathematical expression that evaluates exactly to the target.
  - *Template:* "Target: [Target]. Numbers: [Num 1], [Num 2], [Num 3]. Equation: [Expression]"
  - *Example:* "Target: 24. Numbers: 4, 6, 8. Equation: **6 * (8 - 4)**"
- **GT (Greater-Than):** This task isolates sequential and localized relational tracking, forcing the model to complete a mathematical inequality based on a context year.
  - *Template:* "The movie was released in the year [Year]. It became popular in the year [Target]"
  - *Example:* "The movie was released in the year 1975. It became popular in the year **1976**"
- **SI (Semantic Infilling):** This task requires predicting missing intermediate phrases conditioned on both preceding and subsequent text blocks. Because the missing span must be coherent with context on both sides, SI probes bidirectional coordination and non-sequential integration.
  - *Template:* "[Prefix] [Target span] [Suffix]"
  - *Example:* "The chef chopped the **fresh vegetables** and added them to the boiling soup." (where "fresh vegetables" is the target span constrained by both the preceding action and the subsequent context).

Dream-based conditional-entropy criterion Following Section 3, we operationalize the local–global reasoning criterion using Dream, which yields the most stable and interpretable conditional-entropy estimates among the MDMs we study. For a target token x_t, we compare prefix-only entropy, $H_{\text{Dream}}(x_t \mid x_{<t})$, with full-context entropy excluding the target token, $H_{\text{Dream}}(x_t \mid x_{\setminus t})$. A larger positive gap indicates stronger global reasoning, while a smaller or negative gap indicates a more local or prefix-dominant regime.

Entropy-gap results Table 5 reports Dream-based conditional entropy for the four tasks. COUNTDOWN shows a positive entropy gap, indicating that full-sequence context substantially reduces uncertainty relative to prefix-only conditioning. In contrast, IOI shows a negative gap, consistent with a more local or prefix-dominant regime.

Task	Prefix-only $H_{\text{Dream}}(x_t \mid x_{<t})$	Full-context $H_{\text{Dream}}(x_t \mid x_{\setminus t})$	Gap ΔH
IOI	3.4383 [3.3125, 3.5644]	4.1884 [4.0699, 4.3050]	-0.7500 [-0.9237, -0.5783]
Greater Than	1.4741 [1.4276, 1.5245]	2.2693 [2.1994, 2.3494]	-0.7952 [-0.8836, -0.7133]
Countdown	1.2756 [1.2517, 1.2984]	0.8552 [0.7879, 0.9202]	+0.4204 [0.3456, 0.4911]
Semantic Infilling	4.2733 [4.0007, 4.5420]	3.6705 [3.3333, 4.0050]	+0.6028 [0.2027, 1.0347]

Table 5: Dream-estimated conditional entropy across tasks. Each cell reports the mean entropy, with the 95% confidence interval shown in brackets on the second line. The entropy gap is defined as $\Delta H = H_{\text{Dream}}(x_t \mid x_{<t}) - H_{\text{Dream}}(x_t \mid x_{\setminus t})$. Positive gaps indicate tasks where bidirectional context reduces uncertainty, while non-positive gaps indicate causal or prefix-dominant regimes.

B Detailed Causal Ablation Framework and Results for Countdown

To validate the functional necessity of the front-loaded computation observed in Masked Diffusion Models (MDMs) during the Countdown task, we conduct a targeted causal ablation study on the early-layer components (Layers 2–8). This experiment directly addresses potential artifactual biases introduced by projection-based interpretation tools like the logit lens.

B.1 Experimental Setup

We utilize the rank-based activation masks extracted via the protocol described in Section 3. Instead of merely observing activation values, we systematically ablate the top-ranked task-specific attention heads and MLP dimensions by overwriting their outputs with zero vectors during the forward pass. As a control baseline, we perform an identical ablation procedure on randomly selected dimensions and middle-to-late layer groups (Layers 16–24) across 5 independent random seeds.

B.2 Quantitative Results

The macro-level performance metrics following the causal interventions demonstrate a sharp architectural divergence between the models:

- **Early-Layer Target Ablation:** Disabling the core early-layer circuits in the MDM results in a catastrophic performance drop, with the final task completion accuracy collapsing from an unablated 91.2% down to **35.2%**.
- **Random Control Ablation:** Conversely, ablating an equivalent number of random dimensions or late-layer components yields negligible degradation, with the MDM maintaining a robust accuracy of **89.4%**.

This severe asymmetry in performance degradation provides unambiguous causal evidence that the early layers of MDMs are not just exhibiting passive alignment, but are actively executing the core algorithmic logic required for global planning tasks.

C Experimental Details and Additional Results

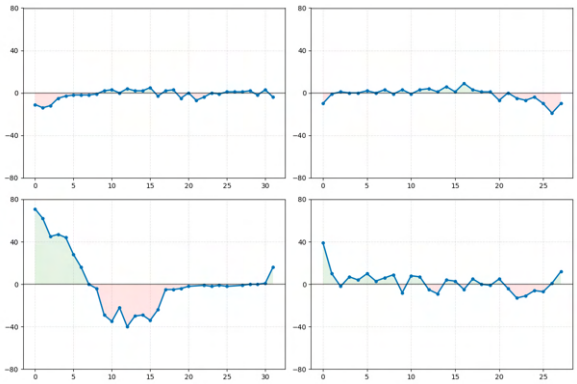


Figure 3: Layer-wise difference in unique attention component usage (MDM−ARM). Rows correspond to IOI and COUNTDOWN; columns compare DiffuLLaMA vs. LLaMA-2 and Dream vs. Qwen. Green indicates greater usage in MDMs, and red indicates greater usage in ARMs.

Computational Resources All experiments were conducted on NVIDIA A6000 GPUs. Circuit extraction and analysis required less than 10 GPU-hours per model. No additional pre-training was performed.

Implementation Details We use HuggingFace Transformers for model loading and inference. Circuit discovery is implemented with Edge Attribution Patching with Integrated Gradients (EAP-IG). We select the top 1,000 edges, corresponding to a faithfulness score of 0.6, as a trade-off between sparsity, attribution coverage, and graph connectivity. In practice, reducing the threshold below 1,000 leads to fragmented circuits: we observe that at 800 edges, the discovered subgraph no longer preserves end-to-end connectivity. By contrast, increasing the threshold yields graphs that more closely approach the full computation graph without providing additional interpretive benefit. The choice of 1,000 edges therefore lies in a stable regime where circuit topology, connectivity, and component rankings remain consistent.

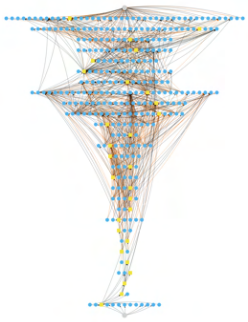
Attribution scores are then aggregated at the component level to identify the Top-K components. We set $K = 100$, the smallest value for which the selected components form a well-connected and interpretable subgraph while preserving end-to-end connectivity. Importantly, these top 100 components account for more than 65% of the total attribution mass, indicating that a relatively small subset of components captures most of the task-relevant computation. Logit-lens projections follow the standard unembedding-based formulation. All other parameters use default library settings.

Licenses and Terms of Use All pretrained models and tools used in this work are publicly released research artifacts. We use them solely for research and analysis in accordance with their respective licenses, and do not redistribute any models or derived data.

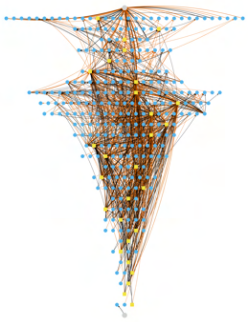
Step-wise Circuit Stability For step-wise circuit extraction, we compute circuits at each diffusion step and aggregate attribution scores across steps. The set of participating components (attention heads and MLP layers) remains largely stable over diffusion time, with more than 70% overlap in selected components between steps. Accordingly, the visualizations in Figures 7 and 6 represent averaged structures; step-wise variation is reflected mainly in the attributed edges rather than in component identity. This suggests that masked diffusion primarily refines information routing within a largely fixed component set, rather than progressively recruiting new components.

Task	Model	Layer	Explanation	Count
IOI	LLaMA	0–4	descriptive adjectives	2821
			expressions of uncertainty or doubt	104
			synonyms for happiness or joy	50
	DiffuLLaMA	0–4	synonyms for happiness or joy	193
			expressions of uncertainty or doubt	80
			technical jargon or specialized terminology	57
COUNTDOWN	LLaMA	0–4	numerical values or quantities	1012
			terms associated with technology and innovation	226
			technical terms related to technology	86
	DiffuLLaMA	0–4	synonyms for happiness or joy	207
			expressions of uncertainty or doubt	114
			terms associated with technology and innovation	72

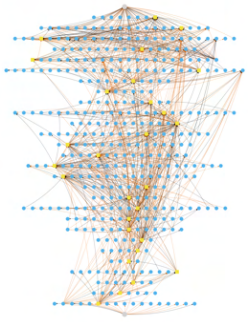
Table 6: Top 3 most common semantic explanations for active neurons in early layers (0–4). Autoregressive models (LLaMA) display sharp specialization, dedicating an overwhelming 2,821 neurons to descriptive adjectives in IOI and 1,012 neurons to numerical values in Countdown. In contrast, MDMs (DiffuLLaMA) display a flat, task-agnostic profile, utilizing only ∼50–200 active neurons per category for broad concepts regardless of the task.



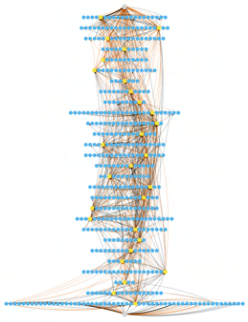
(a) IOI — Qwen-2.5-7B



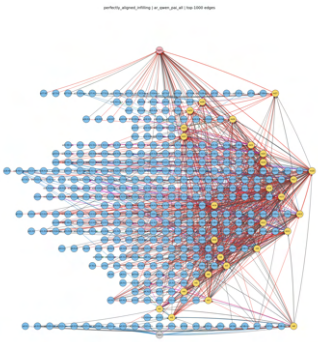
(b) IOI — Dream-Base-7B



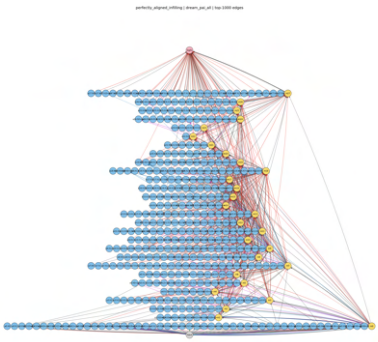
(c) Countdown — Qwen-2.5-7B



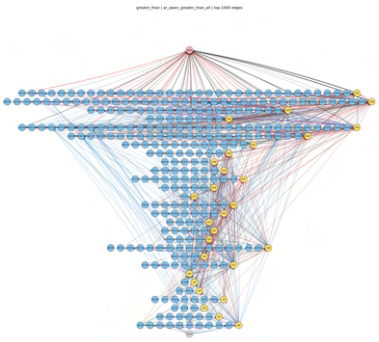
(d) Countdown — Dream-Base-7B



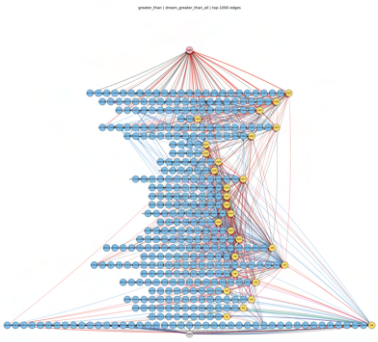
(e) Semantic Infilling — Qwen-2.5-7B



(f) Semantic Infilling — Dream-Base-7B

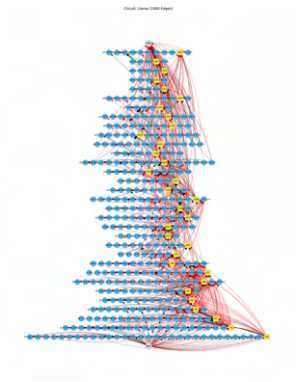


(g) Greater-Than — Qwen-2.5-7B

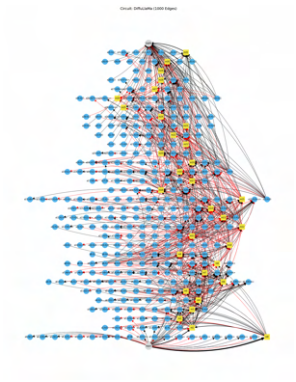


(h) Greater-Than — Dream-Base-7B

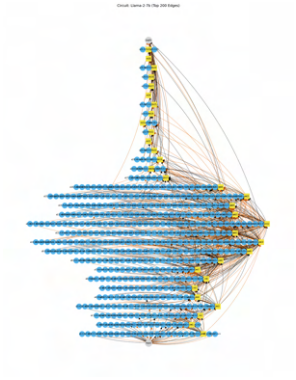
Figure 4: Circuit comparison across tasks and architectures. Left: Autoregressive (Qwen-2.5-7B). Right: Masked Diffusion (Dream-Base-7B). Rows from top to bottom: IOI, COUNTDOWN, SEMANTIC INFILLING, and GREATER-THAN.



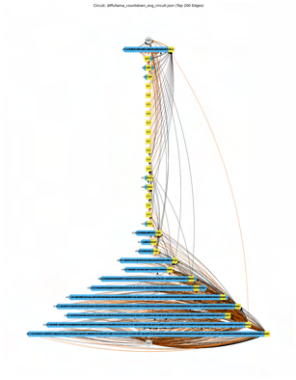
(a) IOI — LLaMA-2-7B



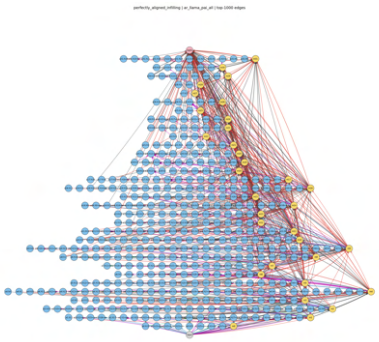
(b) IOI — DiffuLLaMA-7B



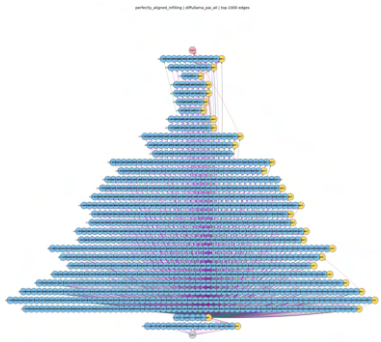
(c) Countdown — LLaMA-2-7B



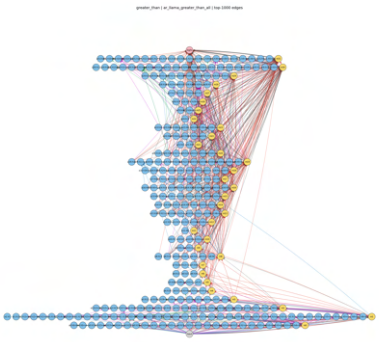
(d) Countdown — DiffuLLaMA-7B



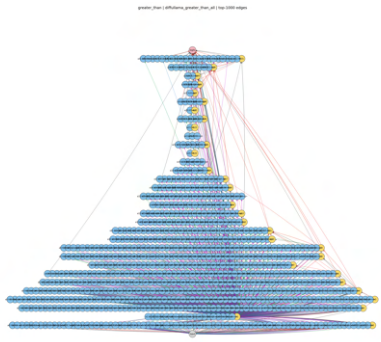
(e) Semantic Infilling — LLaMA-2-7B



(f) Semantic Infilling — DiffuLLaMA-7B



(g) Greater-Than — LLaMA-2-7B



(h) Greater-Than — DiffuLLaMA-7B

Figure 5: Circuit comparison across tasks and architectures. Left: Autoregressive (LLaMA-2-7B). Right: Masked Diffusion (DiffuLLaMA-7B). Rows from top to bottom: IOI, COUNTDOWN, SEMANTIC INFILLING, and GREATER-THAN.

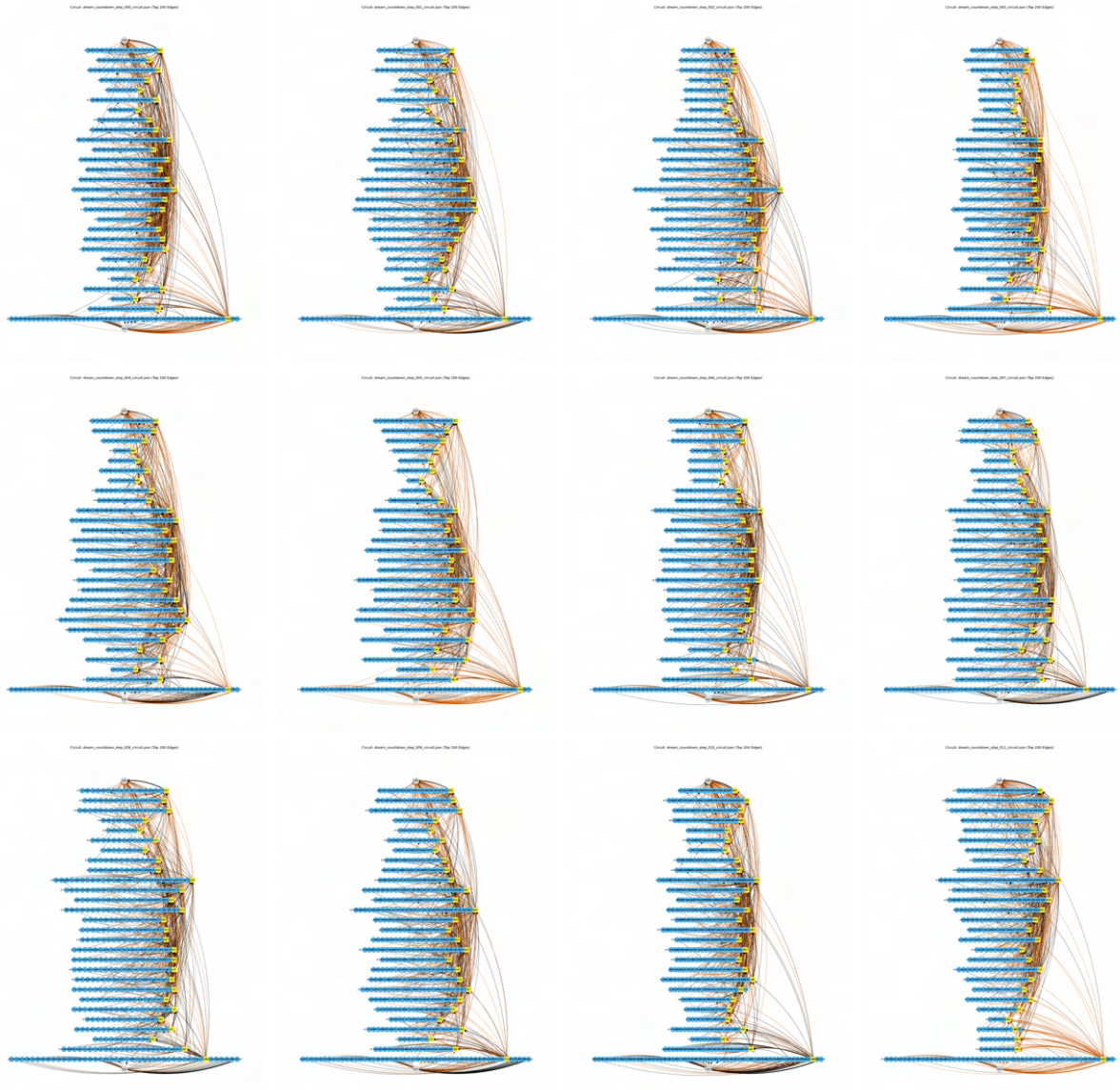


Figure 6: Step-wise circuit visualization of Dream on the COUNTDOWN task. Steps 1–12 are shown from left to right and top to bottom.

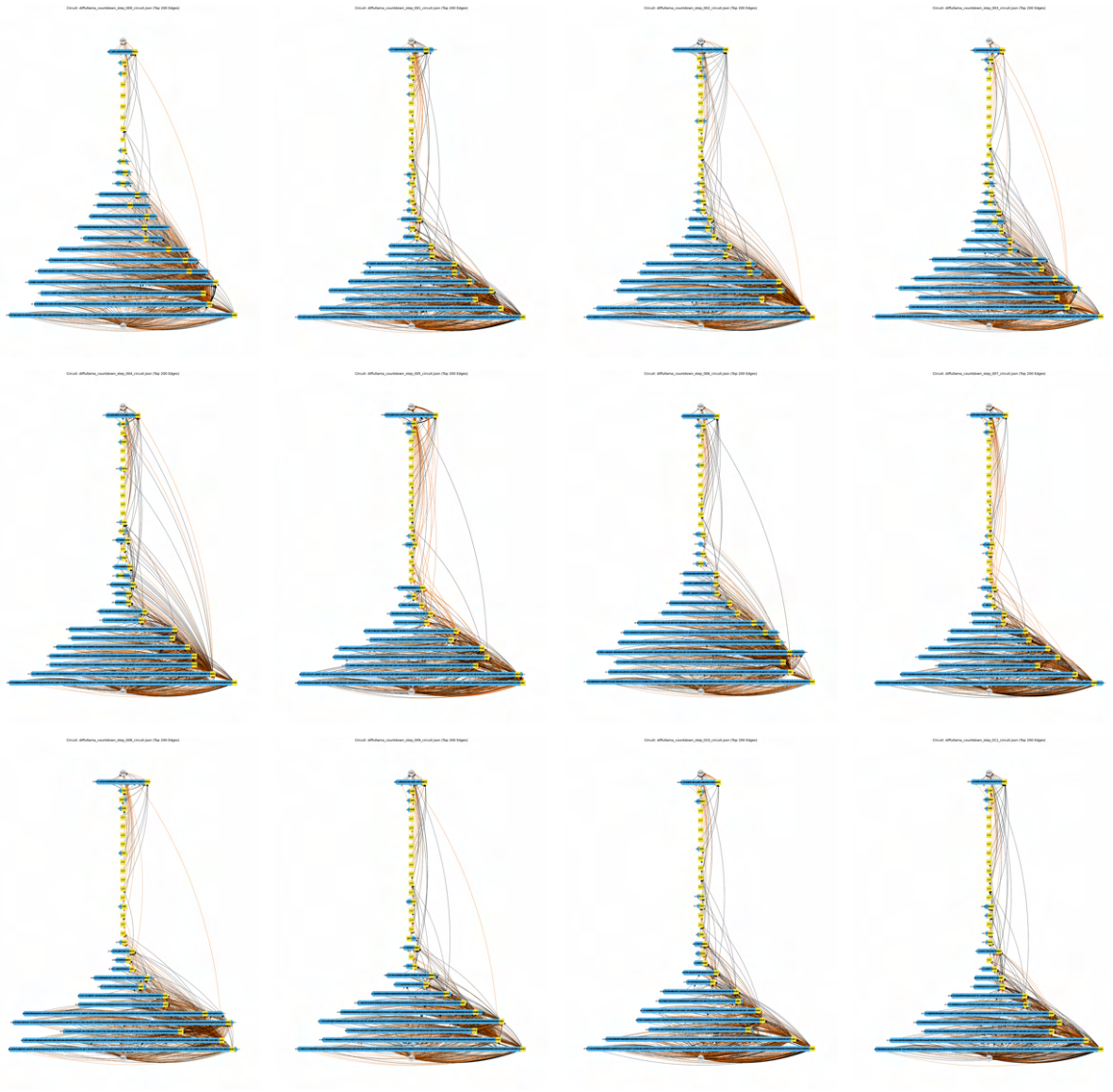


Figure 7: Step-wise circuit visualization of DiffuLLaMA on the COUNTDOWN task. Steps 1–12 are shown from left to right and top to bottom.

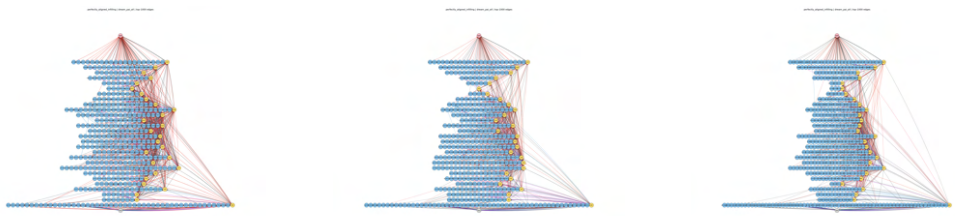


Figure 8: Step-wise circuit visualization of Dream-Base-7B on the Semantic Infilling task. The three panels display the extracted task circuits across its generation intervals from left to right.

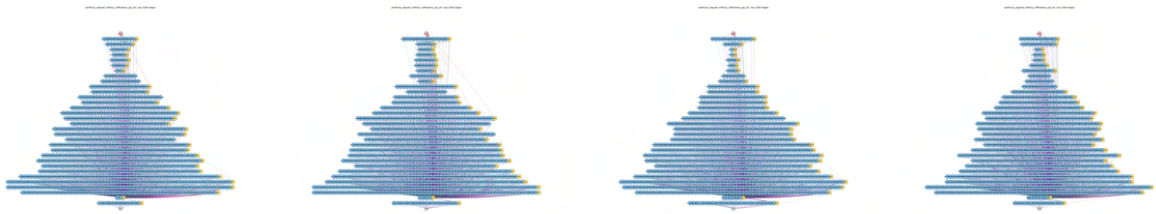


Figure 9: Step-wise circuit visualization of DiffuLLaMA-7B on the Semantic Infilling task. The four panels display the extracted task circuits across its generation intervals from left to right.

Table 7: Top 30 Source Components for **IOI** Task

Model	**Top Source Components (Edges)**
LLaMA-2	input→m0, input→m1, a3.h26→m3, a26.h21→logits, m1→a3.h26$< q >$, a1.h18→m1, input→a3.h26$< q >$, a25.h0→logits, m27→logits, a27.h29→logits, m0→m1, m1→a3.h26$< k >$, a5.h15→a6.h30$< k >$, input→a3.h26$< k >$, a24.h3→logits, a23.h20→logits, a21.h30→logits, a1.h1→m1, m2→a3.h26$< k >$, m0→a3.h26$< q >$, a18.h9→logits, a21.h1→logits, m2→a3.h26$< q >$, m4→a6.h30$< q >$, m0→a3.h26$< k >$, m0→a1.h18$< q >$, input→a1.h18$< q >$, a1.h1→a3.h26$< q >$, a24.h15→logits, a20.h8→logits
Qwen	a24.h24→logits, a26.h15→logits, a23.h11→logits, a27.h21→logits, a27.h4→logits, m27→logits, a27.h1→logits, a26.h26→logits, a26.h22→logits, m20→m22, a27.h18→logits, a27.h17→logits, a27.h5→logits, a27.h14→logits, a24.h23→logits, m24→logits, input→a0.h10$< v >$, a27.h3→logits, a20.h24→m22, a27.h24→logits, a17.h24→a20.h24$< v >$, a26.h5→logits, input→a0.h3$< v >$, a27.h4→m27, m25→logits, a25.h24→logits, a18.h25→a20.h24$< v >$, a18.h27→a20.h24$< v >$, a26.h2→logits, m22→a25.h25$< q >$
DiffuLLaMA	input→m0, input→m1, a3.h26→m3, a28.h7→logits, m31→logits, a26.h21→logits, m0→m1, a1.h18→m1, a27.h29→logits, m1→a3.h26$< q >$, a1.h1→m1, a5.h15→a6.h30$< k >$, m1→a3.h26$< k >$, m1→m4, input→a3.h26$< q >$, input→a3.h26$< k >$, a26.h14→logits, m2→a3.h26$< k >$, a30.h12→logits, m4→a6.h30$< q >$, m0→a3.h26$< q >$, a23.h20→logits, a18.h9→a28.h7$< v >$, input→a1.h18$< q >$, m0→a1.h18$< q >$, input→m4, m2→a3.h26$< q >$, a22.h19→logits, input→m2, a27.h29→a28.h7$< v >$
Dream	a24.h24→logits, m19→m20, a23.h10→logits, m18→m19, m19→m21, m9→m19, m21→m22, a18.h25→m20, m14→m15, m25→logits, m12→m18, m7→m17, a15.h20→m16, m9→m20, m10→m20, a15.h20→m20, m12→m15, m12→m13, m18→m22, m17→m20, m14→m22, a15.h20→m19, a15.h23→m16, m8→m17, m11→m16, m21→logits, a25.h24→logits, a18.h25→m21, m8→m20, m11→m20

Table 8: Top 30 Source Components for **COUNTDOWN** Task

Model	**Top Source Components (Edges)**
LLaMA-2	input→a0.h15< v >, m10→m11, m11→m12, a1.h22→m1, m6→m11, m11→m15, input→a0.h25< v >, m0→a1.h22< v >, m29→logits, m7→m9, m0→m2, m28→logits, m8→m12, m8→a11.h29< v >, m8→m11, a2.h2→m3, m3→m5, m14→m15, m0→m1, input→a0.h3< q >, m13→m15, a12.h5→m13, m7→m10, a12.h22→m12, a5.h15→a7.h6< k >, input→a2.h2< v >, m24→logits, m0→a1.h22< k >, m12→m14, m7→a8.h15< v >
Qwen	m26→logits, m25→logits, m27→logits, input→a0.h3< v >, a23.h11→logits, a25.h12→logits, a26.h22→logits, m24→logits, a26.h24→logits, m26→m27, a24.h23→logits, m21→a23.h11< v >, a0.h3→m0, a22.h13→logits, a26.h23→logits, a23.h11→m25, m25→m27, a23.h19→logits, a25.h12→m27, a26.h25→logits, m21→a25.h12< v >, m20→a23.h11< v >, a26.h26→logits, m25→a26.h22< v >, a23.h11→m26, a26.h22→m27, a26.h22→m26, a26.h24→m27, a23.h11→a26.h22< v >, a24.h23→m25
DiffuLLaMA	m1→m2, input→m0, m1→m3, input→a0.h12< k >, m31→logits, m1→a4.h5< k >, input→a0.h3< k >, input→a0.h15< v >, m1→a3.h3< q >, m1→a3.h27< q >, input→m1, input→a0.h0< k >, input→a0.h3< q >, input→a0.h1< v >, input→a1.h1< v >, m1→a3.h26< k >, m1→a3.h7< k >, m1→a3.h8< k >, m1→m4, m0→m1, m1→a4.h5< q >, input→a0.h13< q >, m1→a2.h2< k >, m0→m3, m1→a3.h0< q >, input→a0.h3< v >, m1→a6.h20< k >, input→a0.h13< v >, m1→a3.h17< k >, m1→a5.h23< q >
Dream	m25→logits, m27→logits, m26→logits, input→a0.h15< q >, a0.h10→m0, input→a0.h3< v >, m24→logits, input→a0.h10< v >, m23→logits, input→a0.h11< q >, input→a0.h15< k >, input→a0.h15< v >, a0.h3→m0, a25.h12→logits, m0→m1, m26→m27, input→a0.h0< v >, input→a0.h11< v >, a27.h11→logits, a26.h22→logits, a26.h25→logits, m22→logits, m21→m27, a0.h15→m0, input→a0.h11< k >, m25→m26, m22→m27, a25.h12→m27, input→a0.h10< k >, a26.h24→logits

Table 9: Top interpretable tokens for high-attribution components (excluding components stated in table 4). Components are sorted by confidence (probability of the top token).

Task	Model	Comp.	Top Tokens (Probability)
Countdown	**DiffuLLaMA**	m1	sierp (0.936), kwiet (0.045), Hinweis (0.004)
		m2	(U+207B) (5.6e-05), nahm (5.6e-05), Hinweis (5.5e-05)
		m3	iftung (4.1e-05), (U+043D) (U+0434) (U+0434) (4.1e-05), iation (4.1e-05)
		input	rd (3.7e-05), iation (3.7e-05), ness (3.7e-05)
		m0	mes (3.5e-05), led (3.5e-05), med (3.5e-05)
	Dream	m25	out (0.012),)) (0.001), from (0.001)
		m26	,- (7.9e-04), A (7.3e-04), S (6.3e-04)
		m22	nothing (8.7e-05), H (8.6e-05), a (7.2e-05)
		m24	int (8.3e-05), i (7.3e-05), commemor (6.5e-05)
		m21	= (3.2e-05), by (3.1e-05), C (3.0e-05)
		a27.h11	doen (2.9e-05), uteur (2.6e-05), retali (2.4e-05)
		a25.h12	7 (2.2e-05), 8 (1.9e-05), 3 (1.7e-05)
		a26.h22	cosy (2.0e-05), W (1.9e-05), -ok (1.8e-05)
		a26.h25	fourth (1.4e-05), five (1.4e-05), IV (1.4e-05)
		a26.h24	ist (1.2e-05), /S (1.2e-05), SIX (1.2e-05)
		m1	. (1.0e-05), 1 (1.0e-05), # (1.0e-05)
		m0	e (9.0e-06), J (9.0e-06), G (9.0e-06)
		a0.h10	what (7.0e-06), that (7.0e-06), is (7.0e-06)
		a0.h15	one (7.0e-06), e (7.0e-06), the (7.0e-06)
		a0.h3	- (7.0e-06), in (7.0e-06), on (7.0e-06)
		input	(7.0e-06), Increment (7.0e-06), 1 (7.0e-06)
	LLaMA-2	m1	sierp (0.896), Unterscheidung (0.074), kwiet (0.027)
		m24	them (0.009), ihnen (0.004), they (0.003)
		m28	– (0.006), , (0.005), - (0.004)
		m14	/- (7.7e-04), -+ (2.2e-04), ỳ (2.0e-04)
		m15	by (2.7e-04), look (2.1e-04), > (1.5e-04)
		m13	Halle (1.7e-04), Hook (1.2e-04), wa (1.1e-04)
		m11	attan (1.4e-04), iore (1.1e-04), (U+0442)(U+043A)(U+0443) (1.0e-04)
		m12	ieder (1.1e-04), ikai (1.1e-04), sail (1.1e-04)
		m7	bek (9.7e-05), untime (7.7e-05), mina (7.5e-05)
		m9	opsis (8.7e-05), PO (8.2e-05), zug (7.7e-05)
		m10	ador (8.4e-05), rok (8.3e-05), keit (8.3e-05)
		m8	concrete (8.3e-05), OST (8.1e-05), Chor (8.0e-05)
		m6	idenote (7.2e-05), ischof (7.1e-05), asm (7.1e-05)
		m0	bolds (6.6e-05), sce (6.0e-05), hina (5.7e-05)
		m5	kop (6.6e-05), Ő (6.3e-05), Sug (6.2e-05)
		m2	nobody (6.5e-05), nahm (6.0e-05), everybody (6.0e-05)
		m3	uche (5.2e-05), Chronology (5.1e-05), emer (4.9e-05)
		a12.h5	extension (3.9e-05), oba (3.9e-05), extensions (3.9e-05)
		a5.h15	Campbell (3.8e-05), beskre (3.7e-05), :(U+2009) (3.7e-05)
		input	/- (3.7e-05), igny (3.6e-05), Extern (3.6e-05)
		a1.h22	(U+045A)y (3.4e-05), (U+4E0B) (3.4e-05), unci (3.4e-05)

Continued on next page

Table 9 – continued from previous page

Task	Model	Comp.	Top Tokens (Probability)
		a12.h22	tel (3.4e-05), (3.3e-05), guez (3.3e-05)
		a2.h2	zik (3.2e-05), Muse (3.2e-05), èn (3.2e-05)
	Qwen	m26	(U+6027)(U+4EF7) (1.000), B (1.000), & (0.999)
		m27	Human (1.000), ^K (1.000), derive (0.999)
		m24	(U+62EC) (0.973), (U+5973)(U+6027)(U+670B)(U+53CB) (0.955), .ImageAlign (0.914)
		m21	aeda (0.608), so (0.599), to (0.306)
		a26.h24	A (0.032), A (0.006), (G (0.004)
		a26.h23	make (0.008), (make (0.008), .make (0.006)
		a26.h26	-await (4.0e-04), XMLElement (3.2e-04), etail (2.7e-04)
		m0	fkk (3.5e-04), libertine (2.8e-04), [];\n (2.3e-04)
		a26.h25	(U+81EA)(U+52A8)(U+751F)(U+6210) (3.1e-04), /Dk (2.5e-04), line (2.4e-04)
		a26.h22	(U+5341)(U+56DB) (1.9e-04), (U+5341)(U+4E09) (1.8e-04), (U+80B2)(U+4EBA) (1.6e-04)
		a25.h12	4 (1.5e-04), 5 (1.4e-04), Fifth (1.2e-04)
		a23.h19	Five (7.1e-05), five (5.0e-05), 5 (4.1e-05)
		a24.h23	num (2.5e-05), Gall (2.5e-05), num (2.4e-05)
		a0.h3	teenth (2.4e-05), bénéficie (1.9e-05), Noticed (1.9e-05)
		a23.h11	...";\n (1.3e-05), #ac (1.2e-05), ąż (1.2e-05)
		input	(U+304D)(U+3061)(U+3093) (9.0e-06), (U+4E26)(U+4E14) (9.0e-06), (U+6362)(U+53E5)(U+8BDD) (9.0e-06)
IOI	**DiffuLLaMA**	m1	sierp (0.550), kwiet (0.155), Hinweis (0.032)
		m31	in (0.204), to (0.044), \n (0.039)
		a23.h20	Sarah (3.8e-04), Vir (1.1e-04), sar (1.1e-04)
		a28.h7	V (2.9e-04), K (2.0e-04), Kim (2.0e-04)
		a26.h14	David (1.7e-04), David (1.4e-04), dav (1.2e-04)
		a30.h12	William (1.3e-04), Will (1.2e-04), Will (1.1e-04)
		m4	disambiguation (1.1e-04), printStackTrace (9.8e-05), - (9.5e-05)
		m2	nahm (5.5e-05), - (5.5e-05), Hinweis (5.4e-05)
		a27.h29	urn (5.4e-05), urr (5.3e-05), enis (4.9e-05)
		a18.h9	owo (5.1e-05), ceu (4.3e-05), fen (4.3e-05)
		m3	(U+4F1D) (3.5e-05), tu (3.5e-05), adr (3.5e-05)
		a5.h15	temps (3.4e-05), vend (3.4e-05), cancel (3.4e-05)
		a3.h26	lex (3.3e-05), ongodb (3.3e-05), huvudstaden (3.3e-05)
		input	ô (3.3e-05), Horn (3.3e-05), roid (3.3e-05)
		m0	Einzeln (3.3e-05), (U+800C) (3.3e-05), atri (3.3e-05)
		a1.h1	(U+4EA4) (3.2e-05), gate (3.2e-05), Moc (3.2e-05)
		a1.h18	(3.2e-05), jsp (3.2e-05), epen (3.2e-05)
	Dream	m21	pliers (1.4e-05), Tap (1.4e-05), ynom (1.3e-05)
		m13	doen (1.1e-05), bourgeois (1.1e-05), upholstery (1.0e-05)
		m8	wooded (1.1e-05), curt (1.0e-05), Genius (1.0e-05)

Continued on next page

Table 9 – continued from previous page

Task	Model	Comp.	Top Tokens (Probability)
		m9	melodies (1.1e-05), interpolate (1.0e-05), Infantry (1.0e-05)
		m10	forestry (1.0e-05), rhet (1.0e-05), doen (1.0e-05)
		m15	orestation (1.0e-05), secluded (9.0e-06), cosy (9.0e-06)
		m18	Races (1.0e-05), weets (9.0e-06), ife (9.0e-06)
		m19	adjud (1.0e-05), enchanted (1.0e-05), instantiate (1.0e-05)
		m20	blot (1.0e-05), oval (1.0e-05), blinking (1.0e-05)
		m7	glimps (1.0e-05), seeding (1.0e-05), sadd (1.0e-05)
		m11	Tweet (9.0e-06), Intr (9.0e-06), enchanted (8.0e-06)
		m12	milit (9.0e-06), bourgeois (9.0e-06), slashes (9.0e-06)
		m14	enam (9.0e-06), upholstery (9.0e-06), vener (9.0e-06)
		m16	lan (9.0e-06), part (9.0e-06), ocal (9.0e-06)
		m17	Israelis (9.0e-06), commemor (9.0e-06), driv (9.0e-06)
		a23.h10	(8.0e-06), - (8.0e-06), 0 (8.0e-06)
		a24.h24	i (8.0e-06), in (8.0e-06), on (8.0e-06)
		a15.h20	's (7.0e-06), home (7.0e-06), half (7.0e-06)
		a15.h23	doen (7.0e-06), Packages (7.0e-06), classy (7.0e-06)
		a18.h25	ropy (7.0e-06), liner (7.0e-06), Bio (7.0e-06)
		a25.h24	mailed (7.0e-06), RSS (7.0e-06), masturbating (7.0e-06)
	LLaMA-2	m1	sierp (0.865), Unterscheidung (0.110), kwiet (0.022)
		m27	too (0.394), her (0.042), e (0.031)
		a26.h21	Marian (0.076), Pat (0.008), Anne (0.008)
		a25.h0	Richard (0.050), William (0.035), David (0.033)
		a24.h3	Rosa (0.007), Williams (6.4e-04), Alice (6.1e-04)
		a20.h8	Susan (0.004), sus (4.6e-04), suspect (1.4e-04)
		a23.h20	Sarah (0.004), Vir (0.001), vir (0.001)
		a21.h30	Lee (6.1e-04), Kelly (3.3e-04), ee (1.5e-04)
		m4	vy (2.1e-04), disambiguation (2.1e-04), - (2.0e-04)
		a27.h29	arta (1.3e-04), ML (1.3e-04), ignon (1.3e-04)
		a18.h9	Blue (9.5e-05), cyk (9.2e-05), nja (8.8e-05)
		m0	bolds (6.4e-05), sce (6.0e-05), partiellement (5.6e-05)
		m2	nobody (6.4e-05), nahm (5.9e-05), everybody (5.9e-05)
		m3	ime (5.4e-05), (U+82B1) (5.2e-05), ña (5.1e-05)
		input	ny (4.0e-05), ten (4.0e-05), eral (3.9e-05)
		a5.h15	Chronology (3.7e-05), :// (3.6e-05), Extern (3.6e-05)
		a3.h26	erea (3.6e-05), zął (3.6e-05), Songs (3.6e-05)
		a1.h1	(U+4EA4) (3.4e-05), Indep (3.3e-05), gate (3.3e-05)
		a1.h18	Bek (3.3e-05), arguments (3.3e-05), Millionen (3.3e-05)
	Qwen	m27	Human (1.000), Rossi (0.990), “ (0.978)
		m25	Alexander (1.000), shall (0.986), zá (0.970)
		m24	court (0.983), (U+5973)(U+6027)(U+670B)(U+53CB) (0.955), })();\n (0.941)

Continued on next page

Task	Model	Component	Mean Logit	Top Tokens	Role
GT	Qwen	m26	198.04	2, 1, (U+6027)(U+4EF7)	Digit-related component
	Qwen	m27	143.05	nineteen, Kremlin, 2	General lexical component
	LLaMA	m31	23.97	roughly, Sunday, 1	General lexical component
	LLaMA	m30	17.35	1, (U+042A), I	Proper-noun-related component
	Dream	m27	53.85	doen	General lexical component
	Dream	m26	7.56	doen, 6, 9	Digit-related component
	DiffuLLaMA	m1	70.80	sierp, kwiet, Hinweis	General lexical component
SI	Qwen	m26	176.40	S, (U+6027)(U+4EF7), H	Proper-noun-related component
	Qwen	m27	138.79	pedest, Human, Jeremy	Person-name-related component
	LLaMA	m31	23.13	Tags, Royal, ibile	Person-name-related component
	LLaMA	m30	17.11	and, (U+042A), the	General lexical component
	Dream	m27	31.43	1	Digit-related component
	Dream	m26	9.54	courteous, renewables, chops	General lexical component
	DiffuLLaMA	m1	65.30	sierp, kwiet, Hinweis	General lexical component

Table 10: Representative high-attribution components for GT and SI from the completed Top-100 component-filtered logit-lens runs. Non-ASCII tokens are rendered as Unicode code-point sequences to avoid LaTeX encoding ambiguity. Roles are descriptive labels summarizing the observed top-token patterns rather than definitive functional labels.

Table 9 – continued from previous page

Task	Model	Comp.	Top Tokens (Probability)
		m22	thought (0.958), during (0.921), term (0.914)
		m20	(U+6027)(U+4EF7) (0.893), ",__ (0.247), ynos (0.101)
		a27.h17	Christina (0.444), Jessica (0.339), Crystal (0.330)
		a27.h18	Lisa (0.418), Elizabeth (0.282), Nic (0.228)
		a27.h1	Jamie (0.400), Nathan (0.345), Mary (0.275)
		a27.h21	Amy (0.360), Amber (0.331), Adam (0.331)
		a26.h5	Jesse (0.344), Nich (0.217), Rebecca (0.203)
		a27.h3	Katie (0.305), Ken (0.295), Brittany (0.189)
		a27.h14	Heather (0.252), Steven (0.244), Sean (0.223)
		a27.h24	Scott (0.233), Brad (0.221), Kris (0.220)
		a26.h2	Brad (0.227), Megan (0.194), brand (0.180)
		a27.h4	Mary (0.224), Ben (0.223), Mark (0.212)
		a26.h15	Danielle (0.215), Alicia (0.182), Dustin (0.176)
		a24.h23	John (0.013), Thomas (0.013), Kenneth (0.008)
		a25.h24	ch (0.002), William (0.001), w (0.001)
		a24.h24	Gad (0.001), ogen (9.4e-04), Aqu (8.7e-04)
		a26.h22	.AppSettings (6.5e-04), azt (6.0e-04), .d (4.3e-04)
		a26.h26	(U+0625)(U+0639)(U+062F)(U+0627)(U+062F) (1.9e-04), .TRAILING (1.8e-04), inx (1.8e-04)
		a18.h25	_locator (8.1e-05), ="'. (8.1e-05), .instrument (7.8e-05)
		a20.h24	setChecked (7.5e-05), anmar (6.5e-05), CAF (6.4e-05)
		a18.h27	(U+0623)(U+063A)(U+0644)(U+0628) (4.6e-05), dealloc (4.5e-05), (U+FFFD)(U+FFFD) (4.4e-05)
		a17.h24	.setCharacter (3.0e-05), -urlencoded (2.8e-05), entious (2.7e-05)
		input	(U+304D)(U+3061)(U+3093) (9.0e-06), (U+6362)(U+53E5)(U+8BDD) (9.0e-06), (U+4E26)(U+4E14) (9.0e-06)

Table 11: Component-level logit-lens metrics on GT and SI, aggregated over all Top-100 components per model. NAMEFRAC@10 is the fraction of name-like tokens among the top-10 aligned tokens; LOGITGAP, entropy, and ΔLME are averaged, not median, across components. Across both regimes, MDMs, especially Dream and DiffuLLaMA, show smaller LOGITGAP and higher entropy than Qwen, consistent with the reduced single-component dominance reported in the main text.

Task	Model	NameFrac@10	LogitGap	Entropy	ΔLME
GT	Qwen	0.0642	3.1070	2.1713	-0.1456
	LLaMA	0.1323	0.1058	2.2628	-0.0299
	Dream	0.1449	0.0852	2.2884	-0.0430
	DiffuLLaMA	0.1210	0.0374	2.2722	-0.0510
SI	Qwen	0.0565	3.0503	2.1502	-0.2015
	LLaMA	0.1447	0.1001	2.2670	-0.0342
	Dream	0.1650	0.0423	2.2891	-0.0047
	DiffuLLaMA	0.1450	0.0759	2.2633	-0.1433

Expanded GT/SI logit-lens metrics. The following table aggregates the completed Top-100 component-filtered logit-lens outputs for GT and SI, including both Dream and DiffuLLaMA. Token strings in the representative-token appendix are rendered with Unicode code-point escapes when needed; this quantitative table contains only scalar summaries.

D Quantitative Mechanism Metrics & Results

To formalize specialization and dominance in our component-level logit lens analysis, and to provide objective support for our qualitative interpretations, we introduce several quantitative probes. Let $l_i(c)$ denote the logit assigned by component c to token i, and let $T_K(c)$ denote the set of top-K tokens ranked by logit.

D.1 Semantic Alignment and Dominance Probes

Name Alignment Frequency (NameFrac@K): To quantitatively assess whether semantic alignment extends beyond the top-1 token, we measure the fraction of person-name tokens among the top-K aligned tokens for each component:

$$\text{NameFrac@}K(c) = \frac{|T_K(c) \cap \mathcal{N}|}{|T_K(c)|} \tag{1}$$

where $\mathcal{N}$ denotes the set of person-name tokens, identified using a pretrained BERT-based Named Entity Recognition (NER) model (filtered for the `PERSON` label).

Selective Amplification (Logit Gap): To quantify whether a component strongly favors a single token, we compute the logit gap between the highest and second-highest logits. Because softmax probabilities depend exponentially on logit differences, larger logit gaps indicate stronger selective amplification:

$$\text{LogitGap}(c) = l_{(1)}(c) - l_{(2)}(c) \tag{2}$$

where $l_{(1)}$ and $l_{(2)}$ represent the maximum and second-maximum logits, respectively.

Log-Mean-Exp Dominance Gap (ΔLME): Because frequency alone does not measure predictive dominance, we quantify whether name tokens exert stronger influence when they appear by computing a log-mean-exp dominance gap between name and non-name tokens within the top-K set:

$$\Delta\text{LME}(c) = \text{LME}(T_K(c) \cap \mathcal{N}) - \text{LME}(T_K(c) \setminus \mathcal{N}) \tag{3}$$

where the LME for a subset of tokens S is defined as:

$$\text{LME}(S) = \log\left(\frac{1}{|S|}\sum_{i \in S} \exp(l_i(c))\right) \tag{4}$$

This metric controls for group size and directly measures selective amplification.

Distributional Sharpness (Entropy): To formalize whether semantic alignment is concentrated or distributed, we compute the entropy over the normalized top-K logits:

$$H(c) = -\sum_{i \in T_K(c)} p_i \log p_i, \quad \text{where} \quad p_i = \frac{\exp(l_i(c))}{\sum_{j \in T_K(c)} \exp(l_j(c))} \tag{5}$$

Lower entropy indicates sharper specialization (typical of autoregressive models), while higher entropy indicates more distributed alignment (typical of masked diffusion models).

D.2 Architectural Depth Metric

Center of Gravity (CoG): To provide an objective, scalar measure of the "Mechanism Shift" independent of graph layout and visual density, we calculate the Center of Gravity (CoG). The CoG represents the attribution-weighted average layer index of the discovered circuit, identifying where the core computations are localized:

$$\text{CoG} = \frac{\sum_l l \cdot A_l}{\sum_l A_l} \tag{6}$$

where l is the layer index and A_l is the sum of EAP-IG attribution scores for all components in layer l.

The tables below provide the empirical results derived from the metrics defined above. We observe that autoregressive models exhibit larger logit gaps (e.g., Qwen: 2.12; LLaMA: 0.70) than their masked diffusion counterparts (DiffuLLaMA: 0.49), indicating a stronger concentration of probability mass on a single dominant token. Furthermore, DiffuLLaMA exhibits higher entropy (2.01) than autoregressive models (LLaMA: 1.81; Qwen: 0.97), indicating more distributed semantic alignment. Together, these metrics demonstrate the differences in specialization, selective amplification, and architectural depth between ARMs and MDMs.

Table 12: Component-level dominance, logit gap, and entropy metrics on IOI. In contrast to Table 11, values are reported as medians over Top-100 components, following the protocol used for the IOI analysis. Name Token Proportion (Top-1) is the fraction of components whose top-1 aligned token is a person name. Although MDMs occasionally show a higher raw proportion of name tokens, ARMs exhibit substantially stronger selective amplification, with higher ΔLME and LOGITGAP, and sharper specialization, with lower entropy. Dashes indicate metrics not computed in the original IOI run.

Model	Name Token Proportion (Top-1)	Median ΔLME	Logit Gap	Entropy
Dream	19.7%	0.31	–	–
DiffuLLaMA	25.2%	0.05	0.49	2.01
Qwen	6.0%	0.90	2.12	0.97
LLaMA	13.4%	0.89	0.70	1.81

Table 13: Center of Gravity (CoG) for IOI and Countdown tasks. The sharp drop in CoG for Dream on the Countdown task provides quantitative validation of the mechanism shift (front-loading) into earlier layers during global reasoning.

Task	Model	CoG (Layer Index)	Relative CoG (0–1)
IOI	Qwen2.5-7B	17.528	0.548
	Dream-Base-7B	20.432	0.638
COUNTDOWN	Qwen2.5-7B	16.582	0.518
	Dream-Base-7B	4.856	0.152