

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

Injin Kong* Hyoungjoon Lee*,§ Yohan Jo†

Graduate School of Data Science, Seoul National University
mtkong77@snu.ac.kr, hjoon721@snu.ac.kr, yohan.jo@snu.ac.kr

Abstract

Post-training pretrained autoregressive models (ARMs) into masked diffusion models (MDMs) provides an efficient route to diffusion language modeling, but it remains unclear whether the resulting models reuse inherited autoregressive computation or reorganize it for non-autoregressive generation. We compare two 7B ARM-MDM families across four controlled diagnostic tasks and find a task-dependent mechanism shift. On prefix-dominant tasks, MDMs largely preserve inherited high-attribution pathways or exhibit only modest changes in where computation occurs. On globally constrained tasks, the reorganization is substantially stronger, with task-relevant computation shifting toward earlier layers. This depth-wise pattern persists across prompt re-sampling, circuit budgets, and tested inference budgets, while targeted ablations support the functional importance of the identified structures under the tested intervention protocols. At the component level, diagnostic probes suggest that ARMs rely more strongly on sharply specialized components, whereas MDMs exhibit weaker single-component specialization and more diffuse output-space alignment. Together, these results suggest that diffusion post-training selectively preserves or reorganizes inherited computation according to task structure, rather than uniformly replacing autoregressive mechanisms.

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.

These considerations motivate two empirical questions. First, when an ARM is post-trained into an MDM, which inherited high-attribution pathways remain useful and which are reorganized? Second, when reorganization occurs, does it appear

*Equal contribution.

§Visiting intern from the Department of Biosystems & Biomaterials Science and Engineering, Seoul National University.

†Corresponding author.

primarily in pathway identity, depth-wise localization, or the diagnostic organization of task-critical components? We study these questions in four controlled settings where the target behavior and clean–corrupted contrast are sufficiently well defined for circuit analysis. Our conclusions are therefore scoped to these tasks and two 7B ARM–MDM families, rather than to open-ended generation or diffusion language models generally.

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 pre-trained 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 frac-

tion 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).

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 within-family comparison allows us to characterize computational changes associated with the overall ARM-to-MDM adaptation process. Because the diffusion objective, denoising interface, post-training data, and optimization trajectory are jointly changed during this process, we interpret the observed differences as effects of holistic adaptation rather than attributing them to any single factor.

3.1 Models and Configuration

To test whether the qualitative pattern replicates across model families, we evaluate Qwen2.5-7B (Qwen et al., 2025)/Dream-Base-7B (Ye et al., 2025b) and LLaMA-2-7B (Touvron et al., 2023)/DiffuLLaMA-7B (Gong et al., 2025). Comparisons within each pair control model architecture and initialization. Comparisons of effect magnitude across the two families remain confounded by backbone capability, pretraining history, post-training data, and diffusion-training recipe. We therefore treat cross-family magnitude differences descriptively and base our primary claim on the within-pair pattern reproduced in both families.

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,

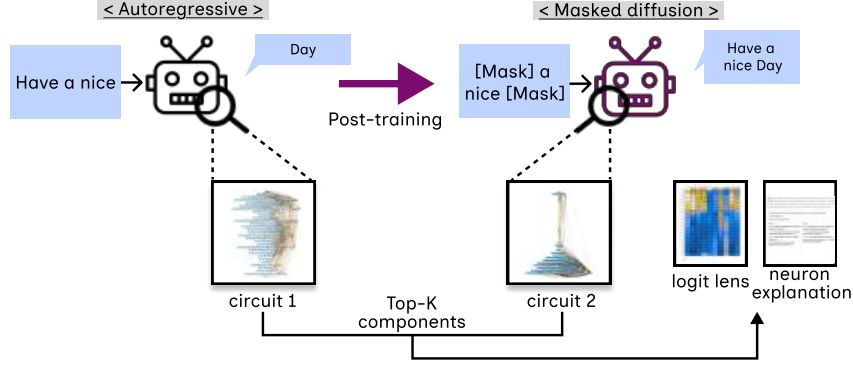


Figure 1: **Overview of the top-attribution-subgraph analysis pipeline.** We extract equal-budget EAP-IG subgraphs for each ARM and post-trained MDM, compare their retained edges, components, and depth-wise attribution, and analyze high-attribution components using diagnostic logit-lens and neuron-level probes. The retained subgraphs are sparse attribution summaries rather than complete task circuits.

whereas in global tasks it depends on constraints across the full sequence. To operationalize this distinction beyond informal task descriptions, we use an MDM-based conditional-entropy gap, $\Delta H := H_{\text{MDM}}(x_t | x_{<t}) - H_{\text{MDM}}(x_t | x_{\setminus t})$, where $x_{\setminus t}$ denotes the sequence with x_t masked. Positive ΔH indicates a global regime, while non-positive ΔH indicates a causal or prefix-dominant regime. We compute the primary grouping with Dream and verify its robustness with LLaDA, which recovers the same sign-based grouping across all four tasks. This entropy criterion is used only to operationalize task grouping and does not enter circuit extraction, overlap, CoG, component-level probes, or ablation analyses. Details are provided in Appendix A.

Behavioral coverage and shared-success protocol. For each task and ARM–MDM pair $p = (A_p, M_p)$, we first evaluate both models on the complete task set $\mathcal{D}$ and identify the pair-specific common-correct pool $\{x \in \mathcal{D} : A_p(x) = y(x) \wedge M_p(x) = y(x)\}$. We then select 500 examples from this pool to form $\mathcal{D}_{\text{circuit}}^{(p)}$, where $y(x)$ denotes the task-defined correct output. All EAP-IG attribution, edge-overlap, component-overlap, Center-of-Gravity (CoG), and downstream component-level analyses are computed exclusively on $\mathcal{D}_{\text{circuit}}^{(p)}$. Consequently, these metrics characterize successful computation on the pair-specific shared-success distribution; they should not be interpreted as describing failure cases or the models’ computation over the complete task distribution.

Indirect Object Identification (IOI): IOI is a canonical circuit-analysis task in which the model

Task	Pair	Acc. (ARM/MDM)	Common-correct n (%)
IOI	Q/D	0.956 / 0.902	603 (86.1%)
	L/DL	0.972 / 0.876	594 (84.9%)
GT	Q/D	0.962 / 0.918	619 (88.4%)
	L/DL	0.943 / 0.832	552 (78.9%)
Countdown	Q/D	0.236 / 0.477	507 (20.3%)
	L/DL	0.253 / 0.322	502 (20.1%)
SI	Q/D	0.632 / 0.753	513 (51.3%)
	L/DL	0.640 / 0.695	511 (51.1%)

Table 1: Behavioral performance and circuit-analysis coverage. Q/D denotes Qwen2.5-7B/Dream-Base-7B and L/DL denotes LLaMA-2-7B/DiffuLLaMA-7B. Acc. reports full-set accuracy for ARM/MDM, respectively. Common-correct n (%) reports the number and proportion of full-set examples correctly solved by both models. We use 500 examples from each common-correct pool for circuit analysis.

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.

3.2.1 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 fi-

nal 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-specific target lengths and prediction settings; details are provided in Appendix A.

3.3 Circuit Discovery and Analysis Pipeline

To trace where and how high-attribution computation changes, we compare matched top-attribution subgraphs in three stages: extraction, attribution-guided comparison, and component-level diagnostics. Extraction ranks edges by EAP-IG attribution and retains an equal-cardinality subgraph for each model. We then compare retained edge identity, influential components, and depth-wise attribution. These sparse subgraphs summarize the highest-attribution computation under a controlled edge budget; they are not assumed to recover the models’ complete task circuits.

3.3.1 Discovery

We use Edge Attribution Patching with Integrated Gradients (EAP-IG) to score edges by their contribution to the task-defined clean–corrupted output contrast. We construct each clean–corrupted pair using the same semantic intervention for the paired ARM and MDM while preserving each model’s native prediction interface. For MDMs, [MASK] tokens are not themselves used as the semantic corruption: the clean and corrupted examples preserve the target positions, mask configuration, denoising schedule, and diffusion-step aggregation, and differ only in the unmasked semantic context that determines the correct output. For ARMs, we apply the same semantic intervention directly to the observed context without introducing mask tokens. Exact task-specific corruption rules are provided in Appendix A.

For each model and task, we retain the 1,000 edges with the greatest absolute attribution, yielding a top-attribution subgraph $\mathcal{S}_{m,t}^{(1000)} \subset \mathcal{G}_m$. The common edge count is a matched-sparsity operating point: it enables ARM–MDM comparisons at equal subgraph cardinality and prevents overlap measures from being directly confounded by differ-

ent retained edge counts. We do not interpret 1,000 edges as a model-specific optimum, a minimally faithful circuit, or a complete task circuit. Sensitivity to equal budgets from 50 to 1,000 edges is reported in Appendix C.1.

3.3.2 Attribution-Guided Circuit Comparison

After extracting a matched-size top-attribution subgraph for each model, we compare whether the MDM retains the same high-attribution edges as its ARM, assigns attribution to different components, or changes the depth-wise distribution of attribution. We conduct these comparisons at the edge, component, and layer levels.

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

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

Table 2: Summary metrics for attribution-guided circuit comparison. Here, $\mathcal{E}^{\text{top}}$ denotes the retained top-attribution edges, $a_{u \rightarrow v}$ denotes the EAP-IG attribution score of edge $u \rightarrow v$, and $A_l = \sum_{v: \ell(v)=l} s(v)$ denotes the total attribution mass assigned to layer l .

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 $z_{\text{AR}}^{(c,t)} = (W_U^{\text{AR}})^\top h_{\text{component}}^{(c,t)}$ for ARMs and $z_{\text{MDM}}^{(c,t,s)} = (W_U^{\text{MDM}})^\top h_{\text{component}}^{(c,t,s)}$ 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 collect the maximally activating input contexts for qualitative and automated explanation, following Bills et al. (2023). 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 3, including the GREATER-THAN(GT) and SEMANTIC INFILLING(SI) evaluations, the results reveal a task-dependent pattern of circuit reuse and reorganization.

Uncertainty across prompt resampling. Paired bootstrap resampling preserves the qualitative depth pattern across tasks. The ΔCoG intervals remain strongly negative for COUNTDOWN (−11.7 [−12.2, −11.1] for Q/D; −11.8 [−12.4, −11.2] for L/DL) and SI (−9.9 [−10.8, −9.0]; −9.2 [−10.1, −8.3]), while IOI shows no early-layer relocation (+2.9 [+2.4, +3.5]; +1.6 [+1.0, +2.1]). This indicates that the qualitative depth-localization pattern remains stable across prompt resampling.

Depth-level sensitivity to the edge budget. We recomputed the comparisons using equal-size top- n attribution subgraphs for $n \in \{50, 100, \dots, 1000\}$.

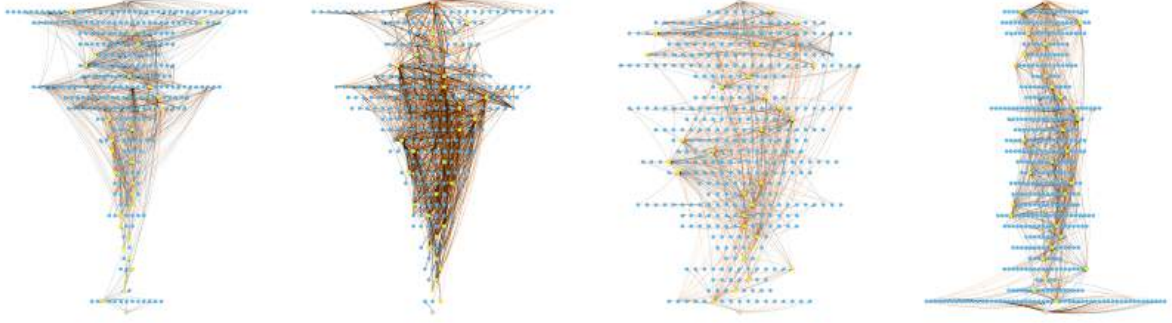


Figure 2: Matched top-1,000 attribution subgraphs for IOI and Countdown. From left to right: Qwen2.5-7B and Dream-Base-7B on IOI, followed by Qwen2.5-7B and Dream-Base-7B on Countdown. Node and edge density show the retained high-attribution structure under a common sparsity budget, not complete task circuits. Higher-resolution versions and remaining tasks are in Appendix Figure 4.

Task	Pair	Overlap		CoG	
		Edge	Top- K	ARM	MDM
Causal regime					
IOI	Q/D	.193 [.174,.214]	.105 [.091,.121]	17.5 [17.2,17.9]	20.4 [20.0,20.9]
	L/DL	.088 [.074,.104]	.124 [.108,.143]	18.2 [17.8,18.6]	19.8 [19.3,20.3]
	Q/D	.290 [.268,.313]	.515 [.489,.542]	18.7 [18.3,19.1]	14.4 [14.0,14.9]
GT	L/DL	.054 [.044,.066]	.117 [.101,.135]	17.3 [16.9,17.8]	11.7 [11.2,12.1]
Global regime					
COUNTDOWN	Q/D	.008 [.005,.012]	.093 [.079,.109]	16.5 [16.1,16.9]	4.8 [4.5,5.1]
	L/DL	.032 [.025,.041]	.081 [.067,.098]	17.1 [16.6,17.5]	5.3 [4.9,5.7]
	Q/D	.225 [.191,.263]	.429 [.386,.471]	13.7 [13.0,14.4]	3.8 [3.4,4.3]
SI	L/DL	.018 [.010,.031]	.124 [.096,.158]	13.1 [12.4,13.8]	3.9 [3.4,4.5]

Table 3: Edge-set similarity and depth-wise Center of Gravity for matched top-1,000 attribution subgraphs. Values are point estimates with 95% bootstrap confidence intervals in brackets. Higher Jaccard overlap indicates greater reuse of retained high-attribution edges, while lower CoG indicates earlier attribution concentration. Q/D denotes Qwen/Dream and L/DL denotes LLaMA/DiffuLLaMA. These metrics describe a matched-sparsity operating point rather than recovery of a complete circuit.

Across this tested range, Countdown and Semantic Infilling retain substantially earlier MDM localization in both model families, Greater-Than retains a smaller earlier shift, and IOI shows no consistent early-layer MDM relocation. The exact retained edges and numerical Jaccard values vary with n , particularly when overlap is close to zero. We therefore interpret this sweep as evidence of depth-level robustness within the tested 50–1,000-edge range, not as threshold invariance of edge identity. We make no robustness claim for budgets above 1,000, below 50, or attribution-mass-based thresh-

olds. Moreover, stability of the depth profile does not imply that any retained subgraph constitutes a complete task circuit. Full results are reported in Appendix C.1.

For IOI, MDMs show substantial preservation of high-attribution structure inherited from their ARM initializations. Table 3 summarizes this pattern using edge overlap and CoG. Although the overlap score is modest (e.g., 0.193 for Qwen/Dream), targeted ablation indicates that the shared edges are functionally important under the tested intervention protocol: 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), consistent with greater reuse of inherited computation rather than a large depth-wise relocation.

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 variation across model families. GT, like IOI, remains closer to the causal regime than the global tasks, through milder depth shifts, higher overlap, or both. SI, like COUNTDOWN, exhibits a

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 4: 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.

strong shift toward earlier layers, consistent with greater reliance on bidirectional context. 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 exhibit the same qualitative split, with greater preservation on causal tasks and stronger early-layer relocation on global tasks. Because the two pairs differ in backbone capability, pretraining history, post-training data, and diffusion-training recipe, we interpret these cross-family differences descriptively rather than attributing them to any single factor.

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.

4.2 Semantic Reorganization of Task-Critical Components and Early Layers

To understand how circuit-level differences translate into 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 high-attribution components align with task-relevant output-space directions and how concentrated these alignments are across components. Second, we analyze *early-layer neuron activations* to characterize aggregate activation patterns in re-

gions where MDMs emphasize computation but component-level output-space alignment appears 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 5 and 11.

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 4, 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 ΔLME values of 0.90 for Qwen and 0.89 for LLaMA, compared with 0.31 for Dream and 0.05 for DiffuLLaMA (Appendix Table 14). 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 5). 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 semantic tokens. Even when circuit topology is similar, diffusion post-training can alter output-space alignment patterns and weaken the dominance of individual name-aligned 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

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 5: 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.

planning is approximated through sequential, component-centric heuristics (Table 5)—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 3). To test whether this early-layer localization is functionally relevant beyond the logit-lens diagnostics, we perform targeted ablations on the identified structures. Ablating these early-layer structures substantially reduces COUNTDOWN accuracy relative to matched random and late-layer controls, providing interventional evidence that they contribute to COUNTDOWN performance under the tested ablation protocol (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 13). 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 suggest a component-level reorganization in the diagnostic output-space patterns induced by diffusion post-training. Whereas ARMs exhibit stronger single-component specialization, MDMs show weaker single-component dominance and more distributed output-space alignment across high-attribution components, accompanied on global tasks by a pronounced shift of attribution 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 COUNTDOWN, where the CoG shift in Table 3 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 acti-

vated by descriptive adjectives and modifier-related tokens (e.g., 2,821 neurons in LLaMA; Appendix Table 8), suggesting these layers encode syntactic features before resolving entity identity. In COUNT-DOWN, 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 ~ 50 –200 active neurons per top category; Appendix Table 8), 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 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 8). This pattern suggests dominance by a limited set of highly specialized units. MDMs instead spread interpretable labels across more units (426) with 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 indicate broader, less concentrated category-level activation patterns.

This component-level reorganization is consis-

tent 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 pattern. These findings suggest that diffusion post-training is associated with weaker component-level specialization and more distributed diagnostic alignment, a pattern that accompanies circuit-level shift toward earlier layers.

5 Conclusion

In this paper, we examined how internal computation changes when pretrained autoregressive models are post-trained into masked diffusion models. Across two ARM-MDM families and four controlled tasks, we find a task-dependent pattern: prefix-dominant tasks preserve more inherited high-attribution structure, whereas globally constrained tasks show stronger reorganization and earlier depth-wise localization. Component-level diagnostics further suggest weaker single-component dominance and more distributed output-space alignment in MDMs.

These findings support a *task-compatibility hypothesis*: inherited autoregressive computation may remain useful for prefix-dominant behaviors, while stronger bidirectional or global coordination may require greater reorganization. We view this as a working explanation rather than a complete theory of ARM-to-MDM adaptation. The results also suggest that, as task-relevant alignment becomes less concentrated in individual components, mechanistic analyses may need to complement localized probes with ensemble-, interaction-, and routing-level views.

Several directions remain open. Our paired-model design captures holistic ARM-to-MDM adaptation but does not disentangle the diffusion objective, denoising interface, post-training data, and optimization trajectory. Future work should separate these factors, extend analysis to naturalistic and open-ended generation, and test settings requiring stronger non-local coordination, including verb-final languages such as Japanese, Korean, Turkish, and Basque. It will also be important to test whether the same task-dependent patterns arise in diffusion language models trained from scratch.

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 a holistic adaptation regime in which the diffusion objective, iterative inference schedule, and post-training data exposure are coupled; our inference-budget sensitivity analysis reduces, but does not fully eliminate, the coupling between task structure and denoising budget. Because the two model families also differ in backbone capability, pretraining history, post-training data, and diffusion-training recipe, cross-family differences in reorganization magnitude cannot be attributed to any single factor. 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.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant (RS-2024-00333484) and by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant (RS-2024-00338140, Development of Learning and Utilization Technology to Reflect Sustainability of Generative Language Models and Up-to-dateness over Time), both funded by the Korean government (MSIT). This work was also supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant (No. RS-2026-25551943, Military-Industry-Academia Cooperation Center (Seoul (Yongsan))) funded by the Korean government (MND), and by the National Supercomputing Center with supercomputing resources including technical support (KSC-2025-CRE-0332).

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.
- 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, Jiayi 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*.
- 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 We evaluate the full behavioral pools before constructing the pair-specific common-correct subsets used for circuit analysis. The full evaluation sets contain 700 examples for IOI and GREATER-THAN, 2,500 for COUNTDOWN, and 1,000 for SEMANTIC INFILLING; from each ARM-MDM common-correct pool, we select 500 examples for circuit analysis. For COUNTDOWN, LLaMA-series models use 13 diffusion steps (13 tokens), while Qwen-series models use 12 diffusion steps (12 tokens). For IOI, we use a single-step setting with one target token. Because the task requires predicting only a single token, additional denoising steps would repeatedly refine the same target position, making the single-step setting the most direct comparison. For GREATER-THAN, we align the diffusion budget to the two-token target continuation and average MDM attribution over 2 diffusion steps. For SEMANTIC INFILLING, the diffusion budget follows 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, forc-

ing 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).

A.1 Clean/Corrupted Intervention Details

Because EAP-IG attribution depends on the semantic corruption defining the clean–corrupted counterfactual pair, we specify the task-specific interventions used in our experiments. Across paired ARMs and MDMs, we apply the same semantic intervention while preserving each model’s native prediction interface. For MDMs, clean and corrupted examples use the same target positions, mask configuration, denoising schedule, and timestep aggregation; only the unmasked semantic context is changed. The [MASK] tokens themselves are therefore not used as the semantic corruption.

IOI. We apply the standard name/role corruption by changing the name assignment that determines the correct indirect object while preserving the sentence structure. For ARMs, EAP-IG is applied directly to the paired observed contexts and scored at the target name position. For MDMs, the same target position and mask configuration are used for the clean and corrupted inputs, with only the unmasked name/role context changed. We use a single diffusion step for the single-token target.

Sensitivity to the IOI corruption. To assess whether the extracted structure depends on the semantic corruption, we additionally evaluate a

harder IOI intervention that swaps the relevant name/role assignment using the other in-context name. Under this corruption, ARM-MDM edge Jaccard is 0.136 at Top-50 and 0.390 at Top-1000, while the ARM and MDM CoGs are 16.21 and 19.10, respectively. Thus, exact edge identity is sensitive to the chosen counterfactual, but the depth-level conclusion remains qualitatively unchanged: IOI still shows no pronounced early-layer relocation in the MDM. This analysis concerns sensitivity to the semantic corruption; sensitivity to alternative MDM masking policies is a separate question.

GREATER-THAN. We change the reference year within the same century (e.g., 1352 $\rightarrow$ 1301) while preserving the target continuation format. For ARMs, EAP-IG is applied to the paired observed prefixes and scored on the target continuation. For MDMs, the same target span, mask realization, diffusion-step budget, and timestep aggregation are used for both inputs; only the reference-year context is changed.

SEMANTIC INFILLING. We replace the constraining context item with a matched semantic contrast while keeping the target infill span fixed. For ARMs, EAP-IG is applied to the paired clean and corrupted contexts and scored on the same target span. For MDMs, the same masked target span, mask realization, denoising schedule, and timestep aggregation are retained, while only the unmasked semantic context is changed.

COUNTDOWN. We shift the target by +1 and shuffle the source-number order, retaining only solvable corrupted instances with the same output-span length. For ARMs, EAP-IG is applied to the paired prompts and scored on the corresponding solution-expression output. For MDMs, the same output-span mask configuration, diffusion-step budget, and timestep aggregation are used for the clean and corrupted inputs; only the target and source-number context is changed.

B Detailed Causal Ablation Framework and Results for Countdown

To test the functional relevance of the front-loaded computation observed in Masked Diffusion Models (MDMs) on COUNTDOWN, we conduct a targeted ablation study on early-layer components (Layers 2–8). This intervention complements the projection-based logit-lens analysis by directly

measuring the performance effect of ablating the identified structures.

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 performance changes under the targeted interventions reveal a clear difference between the identified early-layer structures and the control ablations:

- **Early-Layer Target Ablation:** Ablating the selected early-layer components in the MDM reduces final task completion accuracy from an unablated 91.2% to **35.2%**.
- **Control Ablation:** Ablating an equivalent number of randomly selected dimensions or middle-to-late layer components yields substantially smaller degradation, with accuracy remaining at **89.4%**.

The large difference between targeted and control ablations provides interventional evidence that the selected early-layer structures make a substantial contribution to COUNTDOWN performance under this intervention protocol. We do not interpret this result as establishing a general causal role for early layers across global planning tasks.

C Experimental Details and Additional Results

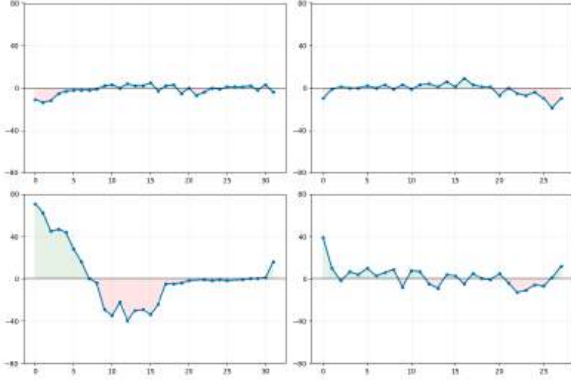


Figure 3: Layer-wise difference in unique attention component usage (MDM-ARM). Rows correspond to IOI and COUNTDOWN; columns compare DiffuLLaMA vs. LLaMA-2 and Dream vs. Qwen. Green indicates greater usage in MDMs, and red indicates greater usage in ARMs.

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). All primary ARM-MDM comparisons retain the top 1,000 edges by absolute EAP-IG attribution. This value is used as a common matched-sparsity operating point, not as a model-specific optimum, minimal faithful size, or estimate of a complete task circuit. Equal-sized subgraphs ensure that Jaccard overlap is not directly confounded by different retained edge counts. We evaluate depth-level sensitivity over $n \in \{50, 100, \dots, 1000\}$ and do not claim robustness outside this range. In particular, the present experiments do not evaluate 2,000-edge subgraphs, arbitrary attribution-mass thresholds, or model-specific iso-faithfulness thresholds.

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.

C.1 Equal-Budget Sensitivity

Table 6 summarizes the equal-budget sweep. Countdown and Semantic Infilling retain strong earlier MDM localization throughout the tested range in both families. Greater-Than retains a smaller earlier shift, whereas IOI exhibits no early-layer MDM relocation. Because exact edge overlap varies with the retained-edge budget, the sweep supports robustness of the qualitative depth profile rather than invariance of edge identity.

C.2 Inference-Budget Sensitivity

To test whether the observed depth relocation can be explained by the number of denoising iterations, we vary only the inference budget within GREATER-THAN and SEMANTIC INFILLING, while keeping the evaluation examples and circuit-extraction procedure fixed.

Across all tested budgets, MDM CoG remains lower than ARM CoG. Moreover, increasing the number of denoising steps does not monotonically increase the CoG gap, and edge overlap changes in different directions across the two tasks. Thus, inference budget alone is insufficient to explain the observed depth relocation, although this analysis does not establish task structure as its sole driver.

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.

Pair	Task	Jaccard range	CV	Depth trend	J_{1000}	CoG _{ARM}	CoG _{MDM}
Q/D	IOI	.193–.227	.048	No consistent shift	.193	17.5	20.4
Q/D	Countdown	.008–.041	.554	Earlier MDM	.008	16.5	4.8
Q/D	Greater-Than	.290–.428	.102	Earlier MDM (mild)	.290	18.7	14.4
Q/D	Semantic Infilling	.176–.288	.116	Earlier MDM	.225	13.7	3.8
L/D	IOI	.088–.123	.101	Later MDM	.088	18.2	19.8
L/D	Countdown	.010–.032	.395	Earlier MDM	.032	17.1	5.3
L/D	Greater-Than	.010–.054	.400	Earlier MDM (mild)	.054	17.3	11.7
L/D	Semantic Infilling	.010–.018	.208	Earlier MDM	.018	13.1	3.9

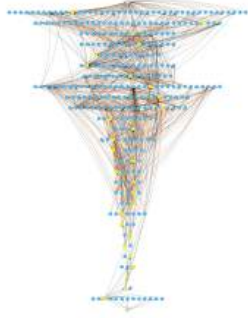
Table 6: Sensitivity across equal-size top- n attribution subgraphs for $n \in \{50, 100, \dots, 1000\}$. Q/D denotes Qwen/Dream and L/D denotes LLaMA/DiffuLLaMA. The qualitative depth profile persists across matched sparsity levels, although exact edge identity varies with the retained-edge budget.

Task	Steps	Edge Ov.	ARM CoG	MDM CoG	Gap
GT	2	.2900	18.65	14.36	4.29
GT	4	.1682	18.65	14.86	3.79
GT	8	.1648	18.65	14.81	3.84
SI	2	.1662	13.68	10.57	3.12
SI	4	.1969	13.68	10.65	3.03

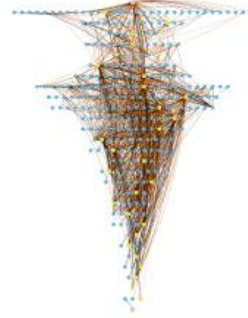
Table 7: Sensitivity to the MDM inference budget. The MDM remains earlier than the ARM at every tested budget, while the magnitude of the depth shift does not increase monotonically with the number of denoising steps.

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

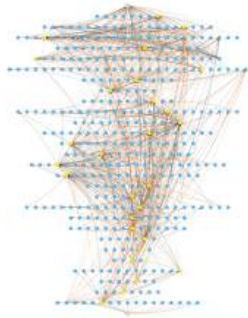
Table 8: Top 3 most common semantic explanations for active neurons in early layers (0–4). Autoregressive models (LLaMA) display sharp specialization, dedicating an overwhelming 2,821 neurons to descriptive adjectives in IOI and 1,012 neurons to numerical values in Countdown. In contrast, MDMs (DiffuLLaMA) display a flat, task-agnostic profile, utilizing only ~ 50 –200 active neurons per category for broad concepts regardless of the task.



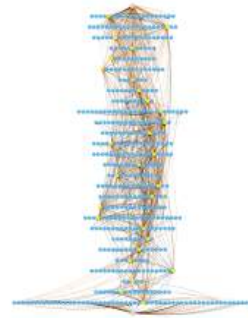
(a) IOI — Qwen-2.5-7B



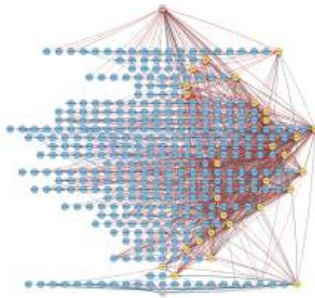
(b) IOI — Dream-Base-7B



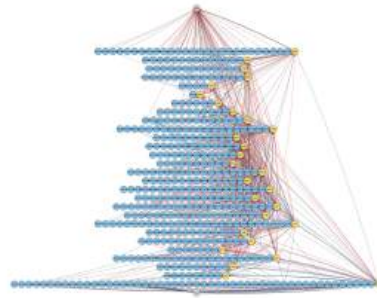
(c) Countdown — Qwen-2.5-7B



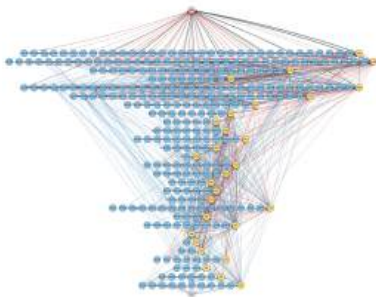
(d) Countdown — Dream-Base-7B



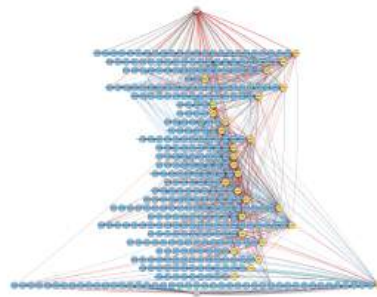
(e) Semantic Infilling — Qwen-2.5-7B



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



(g) Greater-Than — Qwen-2.5-7B



(h) Greater-Than — Dream-Base-7B

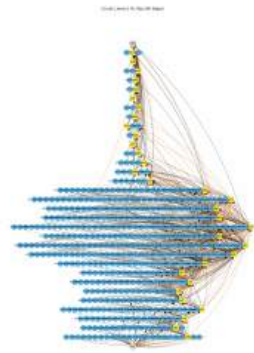
Figure 4: Circuit comparison across tasks and architectures. Left: Autoregressive (Qwen-2.5-7B). Right: Masked Diffusion (Dream-Base-7B). Rows from top to bottom: IOI, COUNTDOWN, SEMANTIC INFILLING, and GREATER-THAN.



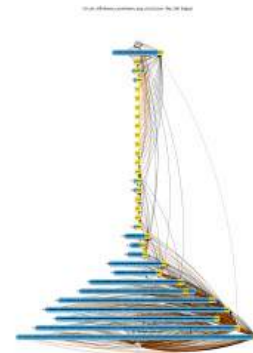
(a) IOI — LLaMA-2-7B



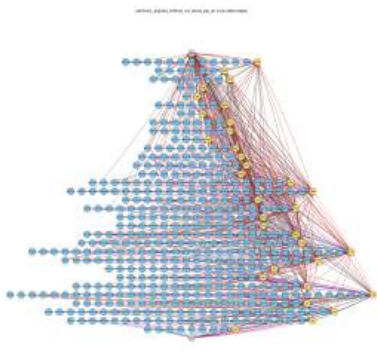
(b) IOI — DiffuLLaMA-7B



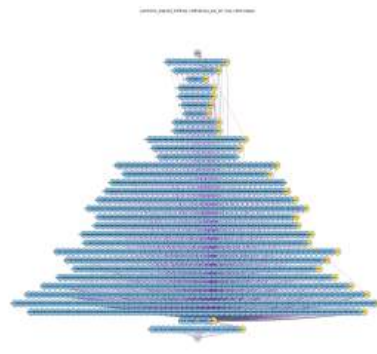
(c) Countdown — LLaMA-2-7B



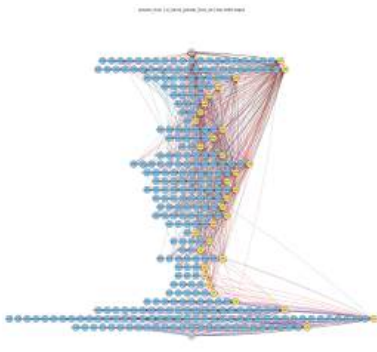
(d) Countdown — DiffuLLaMA-7B



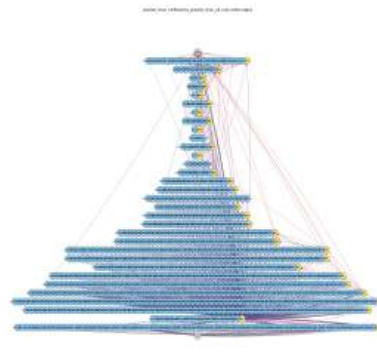
(e) Semantic Infilling — LLaMA-2-7B



(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.

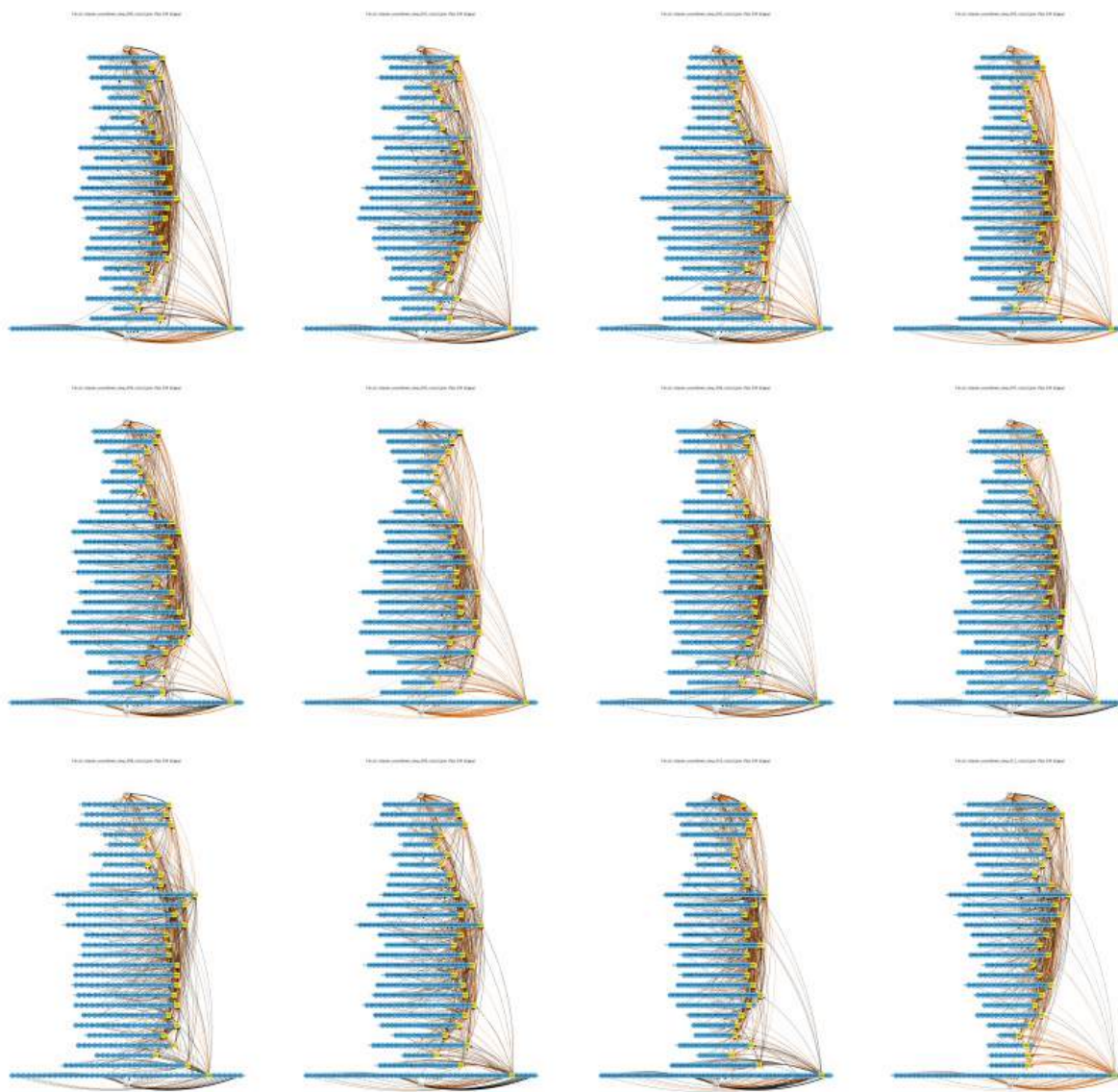


Figure 6: Step-wise circuit visualization of Dream on the COUNTDOWN task. Steps 1–12 are shown from left to right and top to bottom.

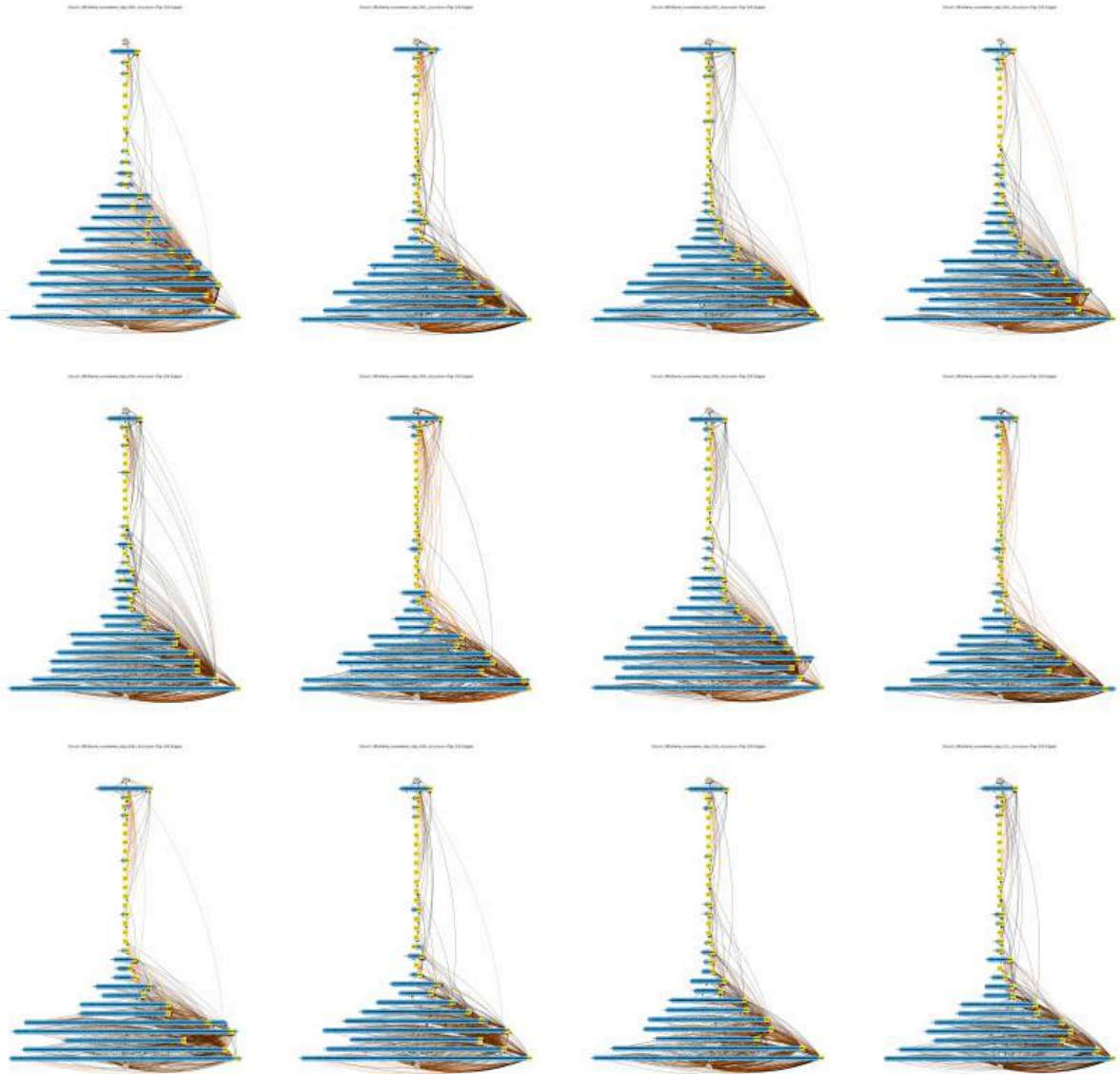


Figure 7: Step-wise circuit visualization of DiffuLLaMA on the COUNTDOWN task. Steps 1–12 are shown from left to right and top to bottom.

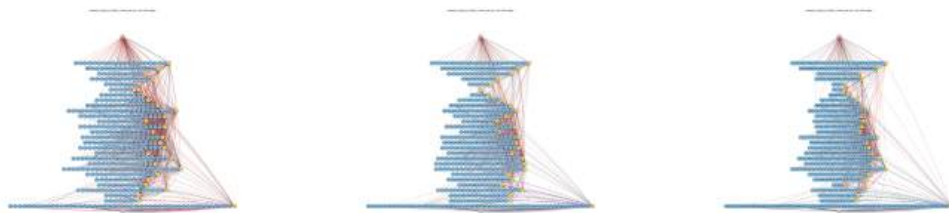


Figure 8: Step-wise circuit visualization of Dream-Base-7B on the Semantic Infilling task. The three panels display the extracted task circuits across its generation intervals from left to right.

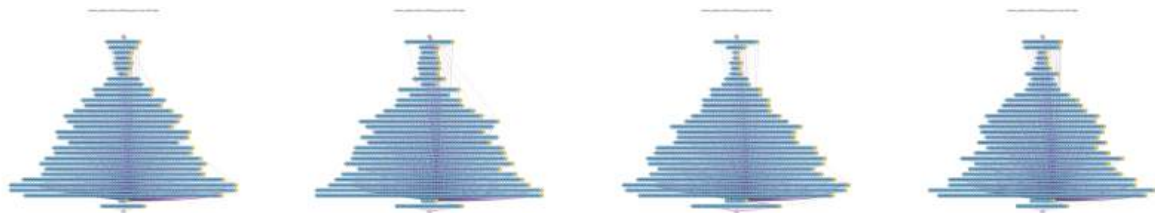


Figure 9: Step-wise circuit visualization of DiffuLLaMA-7B on the Semantic Infilling task. The four panels display the extracted task circuits across its generation intervals from left to right.

Table 9: 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 10: 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 11: Top interpretable tokens for high-attribution components (excluding components stated in table 5). 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)
	LLaMA-2	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)
	LLaMA-2	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)
		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), ÿ (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)
	LLaMA-2	m7	bek (9.7e-05), untime (7.7e-05), mina (7.5e-05)
		m9	opsis (8.7e-05), P0 (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), ů (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)
	LLaMA-2	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 11 – continued from previous page

Task	Model	Comp.	Top Tokens (Probability)
	Qwen	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)
		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), bénéfice (1.9e-05), Noticed (1.9e-05)
		a23.h11	...";\n (1.3e-05), #ac (1.2e-05), aź (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	Einzel (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 11 – 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), ø (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), zqt (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 12: 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 11 – 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 13: 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:

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

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:

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

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:

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

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

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

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:

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

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:

$$\text{CoG} = \frac{\sum_l l \cdot A_l}{\sum_l A_l} \quad (6)$$

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 14: Component-level dominance, logit gap, and entropy metrics on IOI. In contrast to Table 13, 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 15: 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