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

Injin Kong[1,*] **Hyoungjoon Lee**[2,*] **Yohan Jo**[1,†]
[1]Graduate School of Data Science, Seoul National University
[2]Department of Biosystems & Biomaterials Science and Engineering, Seoul National University
mtkong77@snu.ac.kr, hjoon721@snu.ac.kr, yohan.jo@snu.ac.kr

arXiv:2601.14758v3 [cs.LG] 19 Mar 2026

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 the internal algorithmic changes induced by this shift remain poorly understood, leaving it unclear whether post-trained MDMs acquire genuine bidirectional reasoning or merely repackage autoregressive heuristics. We address this question through a comparative circuit analysis of ARMs and their MDM counterparts. Our analysis reveals a systematic "mechanism shift" that depends on the structural nature of the task. MDMs largely preserve autoregressive circuitry for tasks driven by local causal dependencies, but for global planning tasks they abandon initialized pathways and exhibit distinct rewiring with increased early-layer processing. At the semantic level, we observe a transition from sharp, localized specialization in ARMs to distributed integration in MDMs. These findings show that diffusion post-training does not simply adjust model parameters, but reorganizes internal computation to support non-sequential global planning.

1 Introduction

Large language models have achieved near-human performance across diverse linguistic tasks (OpenAI et al., 2024; Qwen et al., 2025; Touvron et al., 2023). Despite these advances, the prevalent autoregressive framework (Radford et al., 2019) imposes structural limitations on generation (Welleck et al., 2019; Bengio et al., 2015). Specifically, sequential generation under causal masking prevents correcting past tokens (Gu et al., 2018; Welleck et al., 2019; Vaswani et al., 2017) so that early errors propagate and amplify throughout the sequence, as the model cannot correct past inaccuracies (Bengio et al., 2015; Ranzato et al., 2016). Moreover, many reasoning and planning tasks require global reasoning, where early decisions must account for constraints that apply to the entire sequence (Gu et al., 2018; Ye et al., 2025a).

Masked diffusion models (MDMs) have gained increasing interest as a non-autoregressive paradigm, with structural properties well-suited to overcoming these limitations (Austin et al., 2021; Sahoo et al., 2024). However, training diffusion-based language models from scratch remains computationally expensive due to slower convergence (Gong et al., 2025). To mitigate this cost, recent work proposes post-training pretrained autoregressive models (ARMs) to the diffusion paradigm (Gong et al., 2025; Ye et al., 2025c). Models such as Dream (Ye et al., 2025c) demonstrate that this strategy can achieve strong performance while requiring only a fraction of the compute needed for training from scratch.

Despite the empirical success of post-training ARMs with diffusion objectives, the specific algorithm changes induced by this post-training process are not yet understood (Gong et al., 2025). It remains unclear whether post-trained MDMs genuinely learn new bidirectional reasoning mechanisms, as intended by the diffusion framework, or instead still rely heavily on autoregressive mechanisms at their core. Although mechanistic interpretability has been applied to understand diffusion models for image generation (Shi et al., 2025; Niedoba et al., 2025), this level of analysis has not yet been extended to text diffusion. Without such analysis, it is difficult to determine if post-trained MDMs genuinely perform global reasoning. If models continue to rely on local, left-to-right heuristics, the theoretical benefits of diffusion-based architectures fail to materialize.

In this work, we address this question by analyzing language models from a circuit-level perspective (Bhaskar et al., 2024), which allows us to directly examine whether diffusion post-training

[*]Equal contribution.
[†]Corresponding author.

induces new computational pathways or primarily reuses existing autoregressive ones. We investigate **where** algorithmic changes occur by comparing circuit structures between ARMs and MDMs post-trained from the same autoregressive backbones, and then examine **how** these changes are realized through detailed analysis using logit lens techniques and neuron-level visualizations.

Through this analysis, we demonstrate that post-training using MDM objectives does not merely alter the training loss but instead induces a systematic reorganization of internal computation—shifting semantic roles across components and revealing a mechanism shift in how language models process and refine linguistic information.

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). This approach allows for non-autoregressive generation using 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. The model is then post-trained to iteratively denoise partially masked inputs, instead of predicting the next token autoregressively. From DiffuLLaMA (Gong et al., 2025) to Dream (Ye et al., 2025c), this line of work shows that post-training pretrained ARMs to MDM objectives can retain many practical advantages of diffusion—such as parallel decoding, iterative refinement, and bidirectional attention—while substantially reducing training cost. Post-trained MDMs also achieve strong performance on directionality-sensitive tasks. However, while these works establish the effectiveness of post-trained MDMs, they largely 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 many behaviors can be explained by sparse circuits involving only 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, further 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.

3 Method

To investigate the mechanistic shift from ARMs to MDMs, we adopt a comparative framework. We analyze pretrained ARMs alongside their directly post-trained MDM counterparts, allowing us to isolate the effect of changing the learning objective from causal modeling to masked diffusion while controlling for architectural confounding.

3.1 Models and Configuration

We conduct experiments across two distinct model families to verify the generalizability of our findings. Specifically, we utilize the Qwen2.5-7B(Qwen et al., 2025) and LLaMA-2-7B(Touvron et al., 2023) architectures.

To enable a direct comparison, we pair each

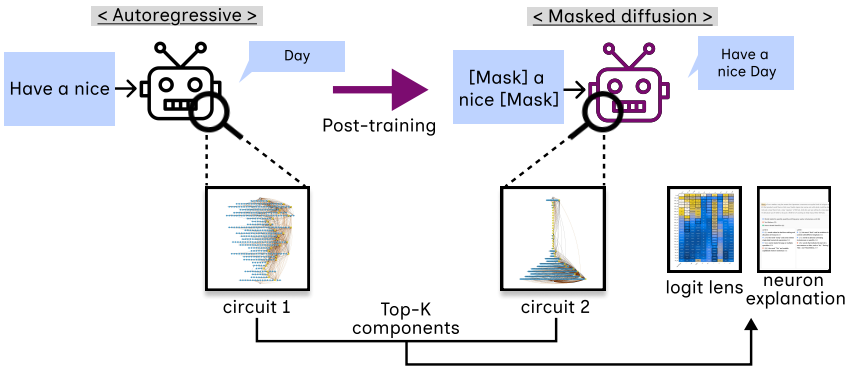


Figure 1: **Overview of the mechanism shift analysis pipeline.** We extract task-specific circuits for both the *ARM* baseline and the post-trained *MDM*. We then identify the *Top-K components* that exhibit the highest topological divergence between the two architectures. Finally, we interpret the algorithmic nature of these shifts using *Logit Lens* and *Neuron Explanation*.

autoregressive checkpoint with its corresponding post-trained MDM derived from the same backbone. Concretely, we compare `Qwen2.5-7B` with `Dream-Base-7B` (Ye et al., 2025c), which is post-trained on the Qwen backbone, and `LLaMA-2-7B` with `DiffuLLaMA-7B` (Gong et al., 2025), which is initialized from LLaMA 2 weights.

3.2 Tasks and Datasets

We select two tasks that lie at opposite ends of the local–global reasoning spectrum: one dominated by autoregressive causal dependencies and one requiring global constraint integration. Let x_t denote the target token at position t, $x_{<t}$ the prefix context, and $x_{\setminus t}$ the full sequence context excluding x_t. Using Dream-based conditional entropy, we define a task as exhibiting more local reasoning when $H_{\text{Dream}}(x_t \mid x_{<t}) \lesssim H_{\text{Dream}}(x_t \mid x_{\setminus t})$, and more global reasoning when $H_{\text{Dream}}(x_t \mid x_{<t}) \gg H_{\text{Dream}}(x_t \mid x_{\setminus t})$. Intuitively, the former indicates that prefix context is largely sufficient, whereas the latter indicates that bidirectional full-sequence context substantially reduces uncertainty. Additional details are provided in Appendix A.

Indirect Object Identification (IOI): A canonical interpretability task primarily solved by Induction Heads in ARMs (Wang et al., 2023). We use this task as a representative benchmark for local reasoning, asking whether induction circuitry—which relies on sequential, left-to-right context to copy tokens—is preserved, dissolved, or transformed when the learning objective shifts from causal prediction to masked diffusion.

Countdown: A numerical reasoning task in which the model is given a set of integers and a target value, and must generate a valid arithmetic expression that equals the target (Ye et al., 2025a). We use this task as a representative benchmark for global reasoning, since it requires inverse planning: decisions made early in the sequence must be conditioned on the final goal. This structure challenges the left-to-right causality of ARMs while favoring the bidirectional context and iterative refinement capabilities of MDMs (Ye et al., 2025b).

Inference Configuration: We use task-dependent generation lengths in all experiments. For IOI, we use a single diffusion step; for COUNTDOWN, we align the number of diffusion steps with the target sequence length. Additional details are provided in Appendix A.

3.3 Circuit Discovery and Analysis Pipeline

We hypothesize that the transition to masked diffusion induces specific algorithmic shifts in tasks where MDMs outperform ARMs (e.g., Countdown). To verify this, we employ a three-stage pipeline: Discovery, Topological Comparison, and Mechanism Interpretation.

3.3.1 Discovery

We employ automated circuit discovery based on Edge Attribution Patching with Integrated Gradients(EAP-IG) (Hanna et al., 2024) to identify the minimal computational subgraph responsible for task performance. For a given task, we identify a sparse subgraph $\mathcal{C} \subset \mathcal{G}$ containing the subset of edges required to keep the model's performance

within a threshold τ of the full model. Within this discovered circuit, nodes represent distinct functional components: attention heads primarily act as information movers that copy content from preceding tokens, while MLPs often serve as associative memories that extract or refine specific semantic attributes from the input state. Consequently, the edges structurally define the algorithmic logic, specifying how these functional outputs are composed and routed to produce the final prediction.

3.3.2 Attribution-Guided Circuit Comparison

To localize *where* algorithmic changes concentrate when transitioning from ARMs to MDMs, we leverage EAP-IG to compare circuits at the level of edges and the components they connect. Rather than treating all discovered edges equally, we focus on those that carry the highest attribution mass under EAP-IG and then identify which components are repeatedly used as sources or sinks of these high-attribution edges.

EAP-IG Edge Overlap: For each model (ARM and MDM), we first select the set of top-attribution edges (1000 edges), $\mathcal{E}^{\text{top}}_{\text{ARM}}$ and $\mathcal{E}^{\text{top}}_{\text{MDM}}$, by ranking edges according to their EAP-IG scores on identical prompts. We quantify overlap using the Jaccard similarity $J(\mathcal{E}^{\text{top}}_{\text{ARM}}, \mathcal{E}^{\text{top}}_{\text{MDM}}) = \frac{|\mathcal{E}^{\text{top}}_{\text{ARM}} \cap \mathcal{E}^{\text{top}}_{\text{MDM}}|}{|\mathcal{E}^{\text{top}}_{\text{ARM}} \cup \mathcal{E}^{\text{top}}_{\text{MDM}}|}$. High overlap suggests that the MDM reuses the same pathways as the ARM, whereas low overlap indicates that diffusion training recruits a distinct set of edges to implement its generative computation. The choice of selecting the top 1000 edges reflects a trade-off between attribution coverage and circuit sparsity and is empirically justified in Appendix B.

Top-K EAP-IG Components: To move from edges to components, we assign each node v a score $s(v)$ by aggregating the EAP-IG scores of all incident edges, defined as the sum of incoming attributions $\sum_{(u,v)\in\mathcal{E}^{\text{top}}} \text{EAP-IG}(u \to v)$ and outgoing attributions $\sum_{(v,u)\in\mathcal{E}^{\text{top}}} \text{EAP-IG}(v \to u)$, where $\mathcal{E}^{\text{top}}$ denotes the set of top-attribution edges for the model. Thus, a component is considered important if it repeatedly appears as either the source or the target of high-scoring EAP-IG edges. We then define the Top-K Components (with $K = 100$) for each model as the nodes with the largest $s(v)$. The choice of K is empirically motivated and discussed in Appendix B.

Layer-wise Center of Gravity (CoG): To quantify where the discovered computation is concentrated across depth, we compute a layer-wise Center of Gravity (CoG) from the EAP-IG attribution profile. Let A_l denote the total EAP-IG attribution mass associated with layer l in the discovered circuit, and define $\text{CoG} = \frac{\sum_l l A_l}{\sum_l A_l}$. Lower CoG indicates more front-loaded computation in earlier layers, while higher CoG indicates greater reliance on middle or late layers. We use CoG as a scalar summary of depth-wise circuit organization. Formal details are provided in Appendix C.

Using these edge-, component-, and layer-level summaries, we first assess the degree of circuit reuse between ARMs and MDMs via overlap of high-attribution edges. High edge overlap provides evidence of mechanistic recycling, where the MDM relies on pre-trained autoregressive pathways, while low overlap indicates the emergence of diffusion-specific generative strategies. Among these edges, Top-K Components pinpoint the nodes at the most influential interactions, yielding a focused set for downstream analysis. CoG then provides a scalar summary of where this computation is depth-wise concentrated, complementing the overlap and component comparisons.

3.3.3 Mechanism Interpretation

Having identified *where* the circuit changes, we investigate *how* the computation differs by applying interpretability techniques specifically to the Top-K EAP-IG Components:

Logit Lens and Component-wise Analysis: Following the logit lens framework (nostalgebraist, 2020), we project intermediate activations into vocabulary space via the unembedding matrix W_U. Since each MDM is obtained by post-training from the same pretrained backbone as its paired ARM, we use W_U as a *shared linear readout basis* across both models: $P_{\text{AR}}(x \mid c, t) = \text{softmax}(W_U h^{(c,t)}_{\text{component}})$, $P_{\text{MDM}}(x \mid c, t, s) = \text{softmax}(W_U h^{(c,t,s)}_{\text{component}})$, where c indexes a component, t the token position, and s the diffusion timestep. We treat these not as the model's native predictive distributions, but as diagnostic probes of which vocabulary directions each component is aligned with under a shared semantic frame.

For autoregressive models (Qwen, LLaMA), we apply the component-wise logit lens to all token positions in the sequence on both IOI and COUNTDOWN tasks. This enables us to trace how the model's representation becomes aligned with the target output across the sequence and to decom-

pose the contributions of the residual stream, attention heads, and MLPs at each token position. For MDMs (Dream, DiffuLLaMA), we examine how these component-wise projections evolve across the diffusion time steps at a fixed set of token positions. Dream supports per-head and per-MLP decompositions, while DiffuLLaMA is analyzed at the level of residual, attention, MLP, and residual-out layers.

This analysis allows us to assess whether a given component has shifted its semantic role—for example, from contributing primarily to next-token prediction in an ARM to encoding a global target or distant operator in an MDM—and how such semantic alignment emerges progressively over diffusion time. Importantly, when applying the unembedding matrix to intermediate component activations, we do not interpret the resulting logits as the model's final predictions. Instead, they serve as a diagnostic probe that reveals which tokens a component is linearly aligned with at a stage of computation.

To complement qualitative inspection with scalar summaries of semantic specialization, we compute quantitative probes from the component-wise logit distributions. Specifically, we measure (i) *NameFrac@K*, the fraction of person-name tokens among the top-K aligned tokens, where person-name tokens are identified using a pretrained BERT-based named entity recognition (NER) model filtered for the `PERSON` label; (ii) *LogitGap*, the gap between the top-1 and top-2 logits, which captures selective amplification; (iii) the *log-mean-exp dominance gap* (ΔLME), which compares the dominance of name versus non-name tokens within the top-K set; and (iv) top-K *entropy*, which quantifies whether alignment is sharply concentrated or broadly distributed. Together, these probes provide an objective characterization of component-level specialization and prediction dominance. Formal definitions and details are provided in Appendix C.

Neuron Explanation: In this work, we define a "neuron" as a single scalar coordinate in the model's residual stream at a given transformer layer (i.e., one element of the hidden dimension of the layer output). For both the IOI and COUNTDOWN settings, we record activations for all combinations of layers, neuron indices, and token positions on the evaluation data. Following the methodology of Bills et al. (2023), we then extract the input tokens that elicit the largest-magnitude activations for a given neuron and use these activating examples for qualitative inspection and automated explanation.

We focus this fine-grained analysis on the Top-K EAP-IG components identified in the previous step: specifically, for high-attribution MLP layers, we visualize the top activating neurons to understand the specific attributes they extract; for high-attribution Attention layers, we analyze the head's output features to determine what information is transmitted. The primary goal of this visualization is to characterize the semantic feature selectivity of the circuit. By mapping high-activation neurons to their corresponding tokens, we aim to determine if MDM components have learned to encode non-causal features (e.g., attending to or encoding "future target" tokens available during the diffusion process) that are fundamentally inaccessible to the ARMs.

Together with the component-level logit-lens probes, this neuron-level analysis helps distinguish whether diffusion post-training changes only the distribution of output-aligned predictions or also reshapes the underlying internal features that support those predictions.

4 Results & Analysis

4.1 Circuit-Level Differences

Figure 2 shows the circuits for ARMs and MDMs on IOI and COUNTDOWN. Combined with the overlap and CoG metrics in Table 1, the results reveal a task-dependent dichotomy.

For IOI, MDMs largely preserve the autoregressive circuitry inherited from their ARM initializations. Table 1 quantifies this structural retention using both edge overlap and CoG. Although the absolute overlap score appears numerically 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 also 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 autoregressive circuitry is no longer sufficient. Circuit overlap drops sharply (e.g., to 0.008 for Qwen/Dream), indicating substantial 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

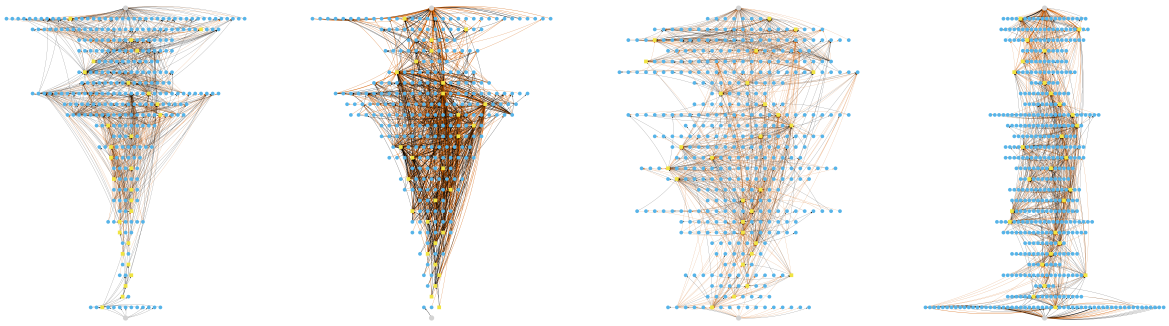


Figure 2: Circuit comparison across tasks and architectures. From left to right: IOI (Qwen2.5-7B), IOI (Dream-Base-7B), COUNTDOWN (Qwen2.5-7B), and COUNTDOWN (Dream-Base-7B). Node labels have been omitted to emphasize the global topological patterns rather than individual components. Blue nodes represent 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 5.

Task & Models	Overlap		CoG (Layer)	
	Edge	Top-K	ARM	MDM
IOI Qwen / Dream	0.193	0.105	17.5	20.4
IOI LLaMA / DiffuLLaMA	0.088	0.124	18.2	19.8
COUNTDOWN Qwen / Dream	0.008	0.093	16.5	4.8
COUNTDOWN LLaMA / DiffuLLaMA	0.032	0.081	17.1	5.3

Table 1: Circuit similarity and Center of Gravity (CoG) shifts between ARMs and their post-trained MDM counterparts. Higher overlap indicates reuse of autoregressive circuitry.

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 2: 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.

usage statistics, which show increased concentration in earlier layers (Appendix B Figure 3).

Taken together, these findings show that masked diffusion post-training does not simply overwrite autoregressive mechanisms. Instead, it preserves causal circuitry where it remains task-compatible, while inducing new early-layer diffusion components when global reasoning exceeds the limits of sequential autoregressive computation.

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 Table 3, 8.

Causal reasoning task (IOI): For IOI, ARMs exhibit *sharply localized semantic specialization*. This pattern appears in the way person-name tokens are concentrated among the aligned outputs of task-critical components. As shown in Table 2, MDMs can place person-name tokens rel-

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 3: 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.

atively often among their top aligned candidates—Dream achieves a higher NameFrac@10 (13.2%) than Qwen (7.9%) and LLaMA (3.4%), while DiffuLLaMA reaches 8.6%—but this does not imply sharper specialization. Instead, ARMs exhibit substantially stronger selective amplification once a relevant name token appears. Qwen and LLaMA achieve median ΔLME values of 0.90 and 0.89, respectively, compared with 0.31 for Dream and 0.05 for DiffuLLaMA (Appendix Table 9). ARMs also show larger logit gaps, while DiffuLLaMA exhibits higher entropy, consistent with a more distributed alignment pattern. In short, raw name frequency does not imply sharper specialization: ARMs remain dominated by a few decisive name-specific components, whereas MDMs spread name-related evidence across a broader set of components.

This qualitative difference is also visible in representative component-level examples (Table 3). In Qwen and LLaMA, several high-attribution attention heads align strongly with person-name tokens, consistent with highly specialized, pointer-like behavior. DiffuLLaMA preserves this pattern, indicating substantial inheritance of autoregressive circuitry. Dream, however, departs from it: its high-attribution components are less clearly name-specific, and their strongest alignments often correspond to non-person proper nouns or broader semantic tokens. Thus, even when circuit topology is similar, diffusion post-training can redistribute semantic roles across components and weaken the dominance of individual name-specific heads.

Global reasoning task (Countdown): A contrasting pattern emerges in COUNTDOWN. ARMs rely on components with *strong numerical selectivity*, sharply aligned with specific digits or operators, suggesting that global planning is approximated through sequential, component-centric heuristics (Table 3). In Dream, this sharp specialization collapses: 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—Dream's CoG on COUNTDOWN drops from 16.5 (Qwen) to 4.8 (Table 1), indicating that diffusion post-training front-loads task-relevant computation toward earlier layers.

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 in Dream by a pronounced shift 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 1 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 5), 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 5), 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 consistent mechanistic pattern. Component-wise logit lens shows how task-relevant outputs are distributed across influential components, whereas neuron-level analysis shows how these representations are implemented in the early layers. More specifically, logit lens probes *semantic alignment at the component level*, revealing which tokens individual attention heads or MLPs are linearly aligned with and how they influence model outputs. Neuron-level analysis, by contrast, characterizes *how semantic information is internally implemented*, independent of direct alignment with the output vocabulary.

In ARMs, explanatory mass is concentrated in a few highly influential components, whereas in MDMs it is distributed across a broader pool of weaker but collectively supportive components. ARMs exhibit a smaller number of unique explanatory components (266) but a higher standard deviation in explanations (369.78), indicating that model behavior is dominated by a limited set of highly influential and specialized components. In contrast, MDMs rely on a larger pool of explanatory components (426) with lower standard deviation (202.52), suggesting that explanatory responsibility is distributed more evenly across components.

ARMs employ a component-centric strategy in which task resolution is dominated by a small number of sharply specialized attention heads or MLPs, supported by task-specific early-layer neurons. MDMs, by contrast, replace these sharp semantic roles with distributed responsibility, activating broader sets of components whose individual contributions are weaker but collectively stable. At the neuron level, this appears as early-layer representations with reduced task-specific selectivity and more uniform activation profiles across tasks.

Taken together, these findings provide a mechanistic account of diffusion post-training. MDMs trade component-level specialization for distributed semantic coverage, supported by early layers that provide broad, task-invariant representational scaffolding. This reorganization explains both the front-loaded computation observed in circuit analyses and the reduced dominance of interpretable components in logit-lens probes, highlighting a fundamental shift from localized causal pathways to more globally integrated computation.

5 Conclusion

In this paper, we investigated where and how masked diffusion post-training reshapes the internal mechanisms of ARMs. To identify **where**, we use circuit analysis and layer-wise CoG to trace changes in circuit composition and depth-wise computation. To understand **how**, we use component-wise logit-lens probing and neuron-level analysis to examine shifts in semantic role assignment and representation patterns. Our results show that masked diffusion post-training induces a task-dependent mechanism shift rather than merely reusing autoregressive computation. In IOI, MDMs largely preserve the core autoregressive circuit while replacing sharp specialization with distributed semantic responsibility. In COUNTDOWN, MDMs show broader circuit rewiring, a stronger shift toward earlier layers, and the same transition to distributed component roles. Overall, masked diffusion post-

training preserves autoregressive mechanisms for local causal reasoning, while reorganizing them for more globally integrated, non-sequential reasoning.

Limitations

Our study is confined to two benchmark tasks, IOI and COUNTDOWN, selected to capture two core yet contrasting modes of computation central to our inquiry: locally causal dependency tracking and global sequence-level planning. This pairing enables a controlled comparison of whether post-trained masked diffusion models preserve autoregressive mechanisms or develop distinct computational circuitry under differing structural demands. Nevertheless, the circuits identified here should be understood as task-specific findings rather than universal signatures of autoregressive or masked diffusion language models. Establishing broader generality will require evaluation across additional tasks spanning diverse compositional, linguistic, and reasoning structures. Furthermore, while our attribution-guided pipeline enables targeted mechanistic analysis by isolating the most behaviorally salient edges and components, it does not constitute an exhaustive characterization of all computations underlying model behavior. A more complete account would necessitate wider circuit sweeps across a broader range of tasks, models, and training conditions — an endeavor that remains computationally prohibitive at scale.

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.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

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.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong. 2025b. 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. 2025c. 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.

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 4 reports Dream-based conditional entropy for the two 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 Δ
Countdown	1.2756 [1.2517, 1.2984]	0.8552 [0.7879, 0.9202]	+0.4204 [0.3456, 0.4911]
IOI	3.4383 [3.3125, 3.5644]	4.1884 [4.0699, 4.3050]	-0.7500 [-0.9237, -0.5783]

Table 4: Dream-based 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 $\Delta = H_{\text{Dream}}(x_t \mid x_{<t}) - H_{\text{Dream}}(x_t \mid x_{\setminus t})$.

B Experimental Details and Additional Results

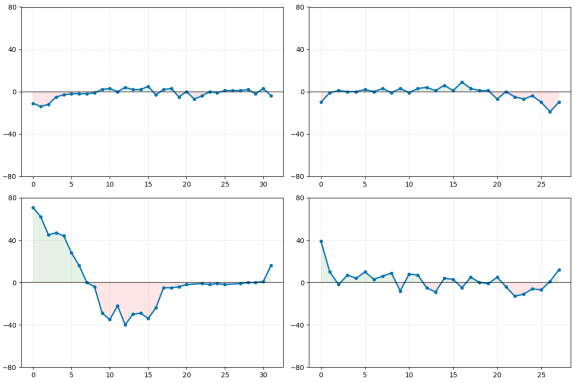


Figure 3: Layer-wise difference in unique attention component usage (MDM−ARM). Rows correspond to IOI and COUNTDOWN; columns compare DiffuLLaMA vs. LLaMA-2 and Dream vs. Qwen. Green indicates greater usage in MDMs, and red indicates greater usage in ARMs.

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

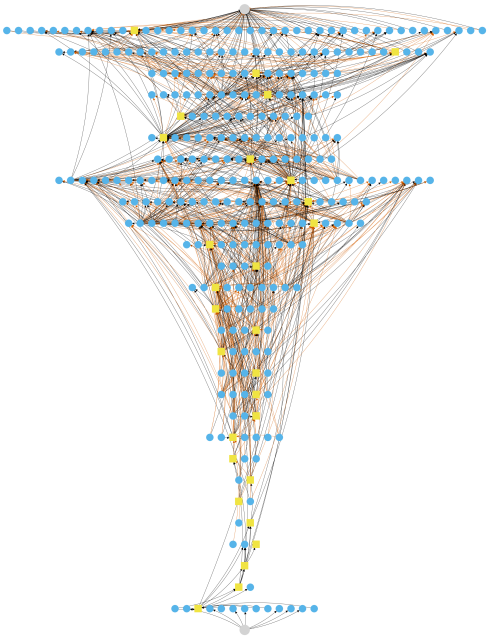
subset of components captures most of the task-relevant computation. Logit-lens projections follow the standard unembedding-based formulation. All other parameters use default library settings.

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.

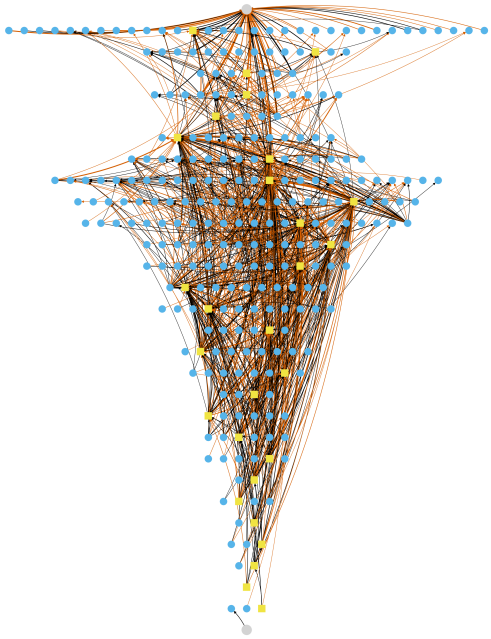
Step-wise Circuit Stability For step-wise circuit extraction, we compute circuits at each diffusion step and aggregate attribution scores across steps. The set of participating components (attention heads and MLP layers) remains largely stable over diffusion time, with more than 70% overlap in selected 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

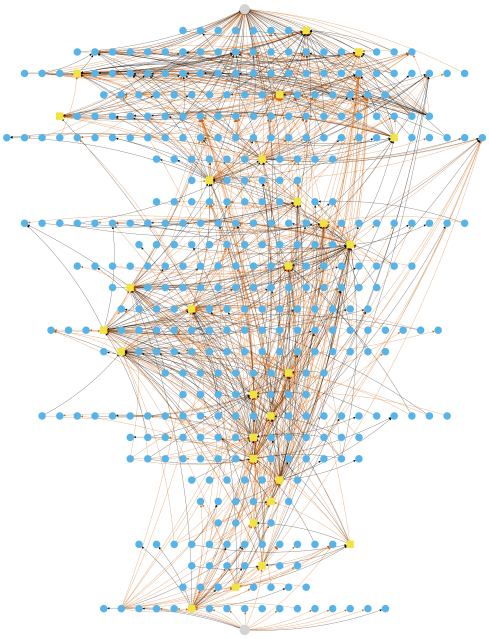
Table 5: Top 3 most common semantic explanations for active neurons in early layers (0–4). Autoregressive models (LLaMA) display sharp specialization, dedicating an overwhelming 2,821 neurons to descriptive adjectives in IOI and 1,012 neurons to numerical values in Countdown. In contrast, MDMs (DiffuLLaMA) display a flat, task-agnostic profile, utilizing only $\sim$50–200 active neurons per category for broad concepts regardless of the task.



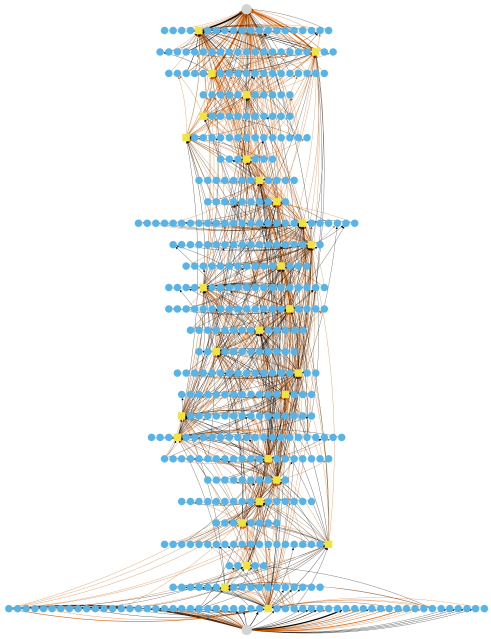
(a) IOI — Qwen-2.5-7B



(b) IOI — Dream-Base-7B

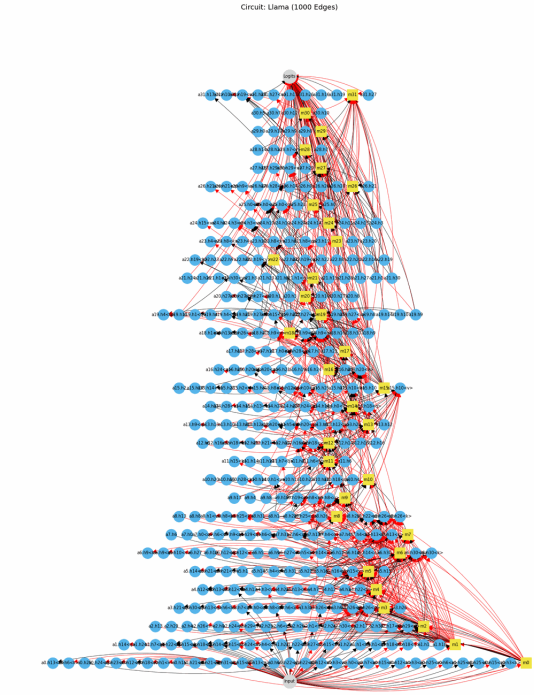


(c) Countdown — Qwen-2.5-7B

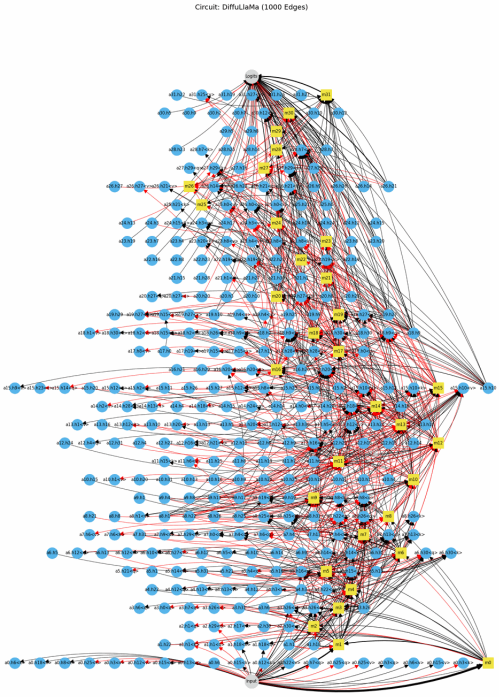


(d) Countdown — Dream-Base-7B

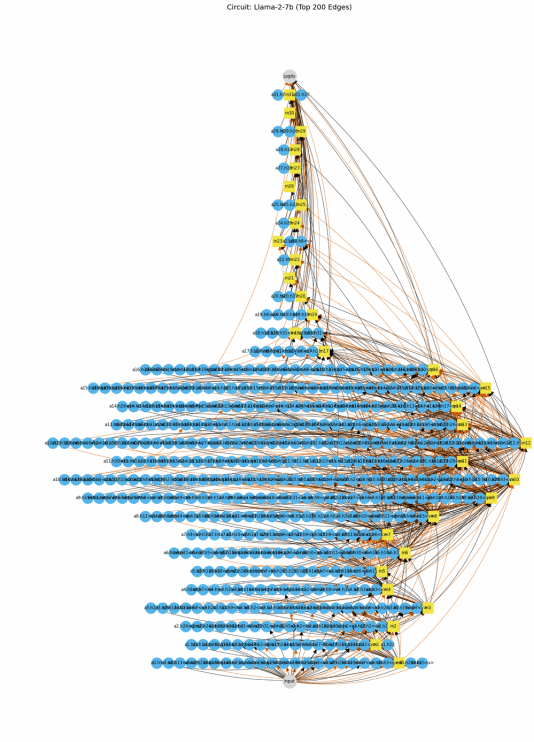
Figure 4: Circuit comparison across tasks and architectures. Top: IOI. Bottom: COUNTDOWN. Left: Autoregressive (Qwen-2.5-7B). Right: Masked Diffusion (Dream-Base-7B).



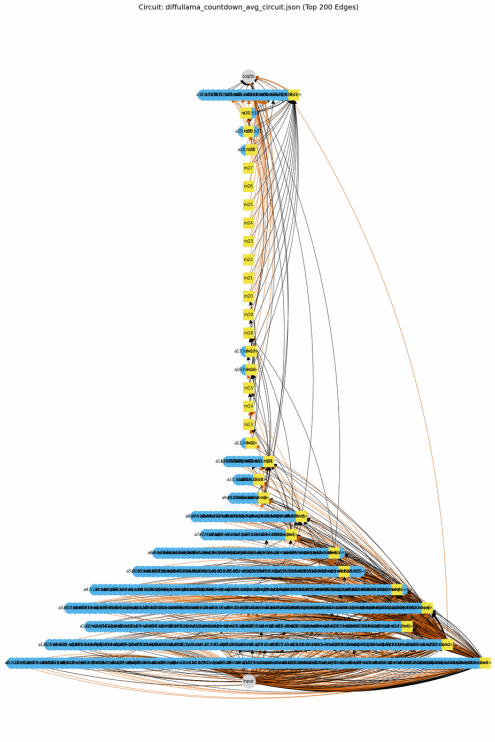
(a) IOI — Llama-2-7B



(b) IOI — DiffuLlaMA-7B



(c) Countdown — LLaMA-2-7B



(d) Countdown — DiffuLLaMA-7B

Figure 5: Circuit comparison across tasks and architectures. Top: IOI. Bottom: COUNTDOWN. Left: Autoregressive (LLaMA-2-7B). Right: Masked Diffusion (DiffuLLaMA).

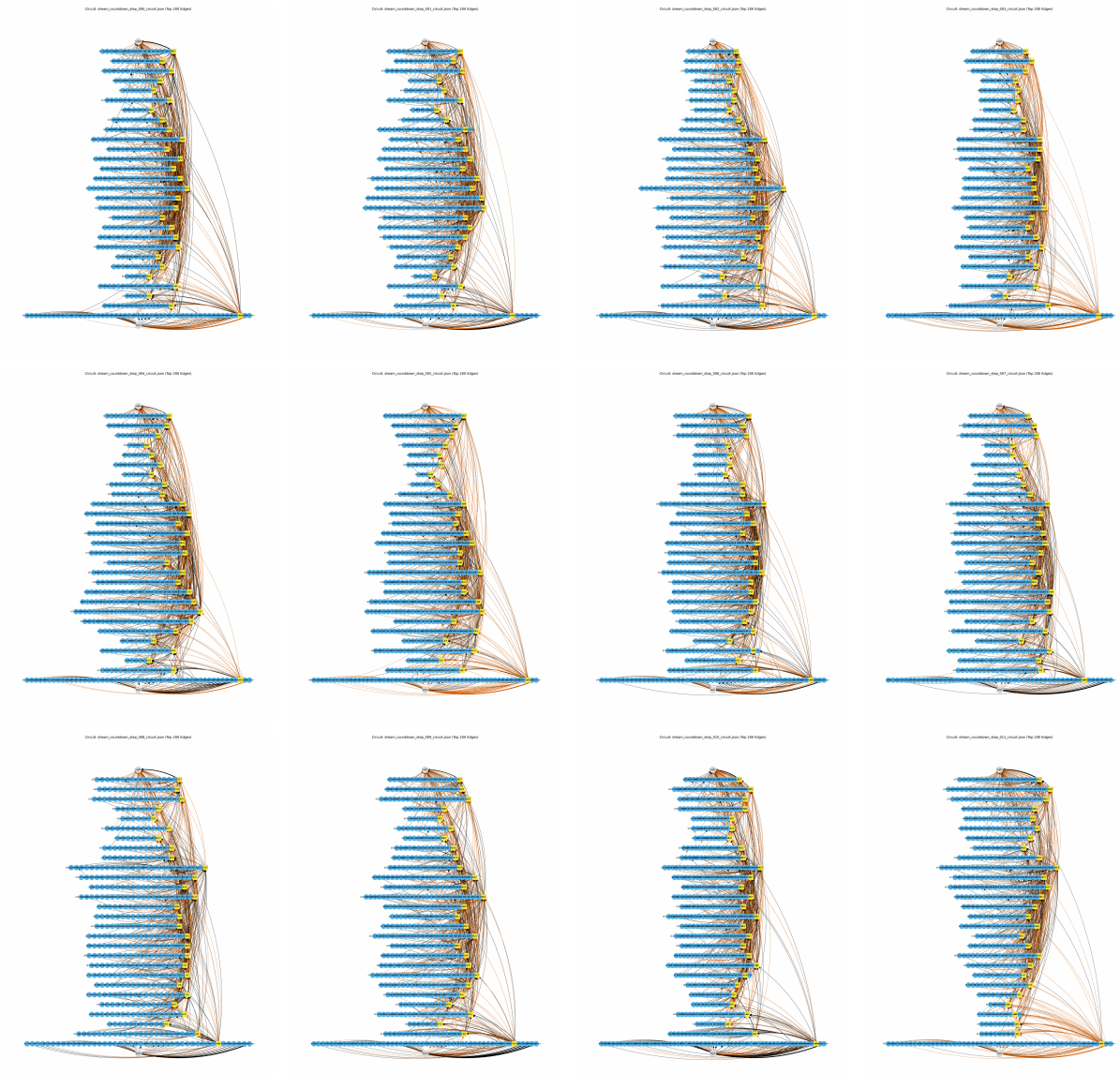


Figure 6: Step-wise circuit visualization of Dream on the COUNTDOWN task. Steps 1–12 are shown from left to right and top to bottom.

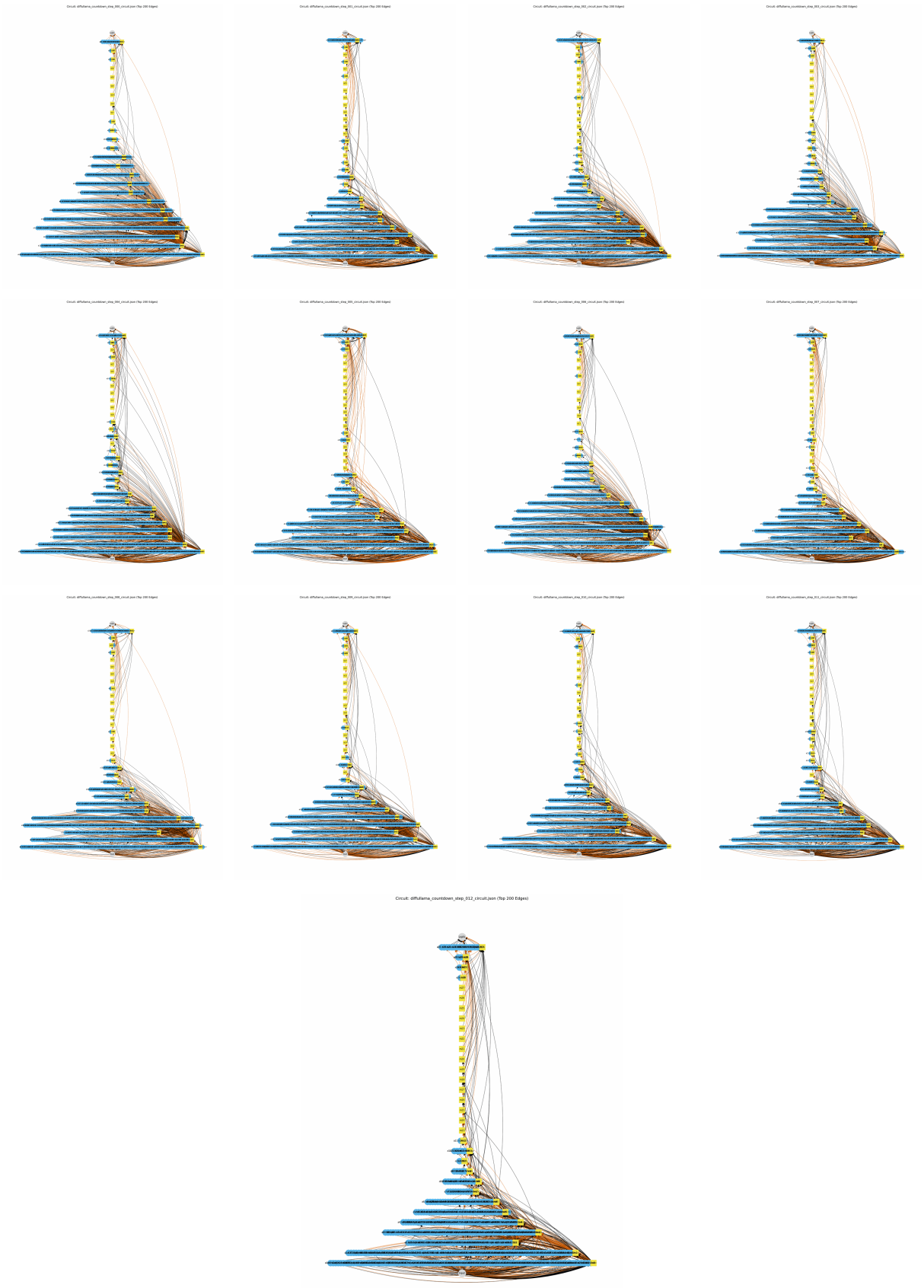


Figure 7: Step-wise circuit visualization of DiffuLLaMA on the COUNTDOWN task. Steps 1–12 are shown from left to right and top to bottom.

Table 6: 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 7: 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 8: Top interpretable tokens for high-attribution components (excluding components stated in table 3). 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 8 – 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 8 – 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

Table 8 – 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)

C 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.

C.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).

C.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 9: Dominance, Logit Gap, and Entropy metrics. While MDMs occasionally exhibit a higher raw proportion of name tokens, ARMs demonstrate substantially stronger selective amplification (higher ΔLME and Logit Gap) and sharper specialization (lower Entropy).

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 10: 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