
Causal Head Gating: A Framework for Interpreting Roles of Attention Heads in Transformers

Andrew J. Nam

Princeton Laboratory for AI
Natural and Artificial Minds
Princeton University
andrewnam@princeton.edu

Henry C. Conklin

Princeton Laboratory for AI
Natural and Artificial Minds
Princeton University
henry.conklin@princeton.edu

Yukang Yang

Department of Electrical and Computer Engineering
Princeton University
yy1325@princeton.edu

Thomas L. Griffiths

Department of Psychology
Princeton University
tomg@princeton.edu

Jonathan D. Cohen*

Princeton Neuroscience Institute
Princeton University
jdc@princeton.edu

Sarah-Jane Leslie*

Department of Philosophy
Center for Statistics and Machine Learning
Princeton University
sjleslie@princeton.edu

Abstract

We present *causal head gating* (CHG), a scalable method for interpreting the functional roles of attention heads in transformer models. CHG learns soft gates over heads and assigns them a causal taxonomy—facilitating, interfering, or irrelevant—based on their impact on task performance. Unlike prior approaches in mechanistic interpretability, which are hypothesis-driven and require prompt templates or target labels, CHG applies directly to any dataset using standard next-token prediction. We evaluate CHG across multiple large language models (LLMs) in the Llama 3 model family and diverse tasks, including syntax, commonsense, and mathematical reasoning, and show that CHG scores yield causal, not merely correlational, insight validated via ablation and causal mediation analyses. We also introduce *contrastive* CHG, a variant that isolates sub-circuits for specific task components. Our findings reveal that LLMs contain multiple sparse task-sufficient sub-circuits, that individual head roles depend on interactions with others (low modularity), and that instruction following and in-context learning rely on separable mechanisms.

1 Introduction

Large language models (LLMs) [1, 2, 3] represent state-of-the-art systems across a wide array of domains, exhibiting remarkable generalization and problem-solving capabilities. Yet, as these models grow in scale and complexity, they become increasingly opaque, making it more difficult to understand, predict, or control their behavior, which raises concerns about safety and misuse [4, 5, 6]. This has motivated a growing body of work on *interpretability*, which seeks to better understand how LLMs learn and represent information, and how their responses can be shaped [7, 8].

*Equal contribution; authors listed alphabetically

Interest has focused in particular on transformer-based architectures [9] such as GPT [1], LLaMA [3], Gemma [10], and DeepSeek [2], in which the central processing blocks consist of multi-head attention followed by multi-layer perceptrons. Here, there has been considerable research on the roles of individual attention heads, which have been found to exhibit some level of human-interpretability [11, 12, 13].

Two broad categories of approaches dominate research on mechanistic interpretability in LLMs. The first uses a trained mapping from latent representations to human-interpretable concepts, such as syntactic features [7, 14, 15] or identifiable items (e.g., the Golden Gate Bridge [16]). The second uses causal interventions to identify portions of a single weight matrix or individual attention heads responsible for a specific behavior [17, 18]. These approaches often focus on small portions of a model, ‘zooming in’ [19] in an effort to interpret the role of a single computational subgraph. However, in deep-learning models, computation is often distributed [20] and the role of one component is dependent on another [21, 22, 23], making the behavior of such complex distributed systems difficult to predict from an understanding of their parts alone [24].

To apply a distributed perspective to mechanistic interpretability, we introduce *causal head gating* (CHG) which identifies a parametrically weighted set of heads that contribute to a model’s execution of a given task. Given a dataset that defines a task, we fit a set of gating values for each attention head that applies a soft ablation to its output using next-token prediction, so that task-facilitating heads remain unaltered while any task-interfering heads are suppressed. Using a simple regularization procedure that further separates irrelevant heads from those that facilitate or interfere with task performance, CHG assigns meaningful scores to each attention head across an entire model according to its task contribution. We use these scores to define a taxonomy of task relevance according to how individual attention heads contribute to a model’s distributed computation of a given task, describing each head as *facilitating*, *interfering* or *irrelevant*. In this respect, CHG offers an exploratory complement to standard hypothesis-driven approaches to mechanistic interpretability, assigning causal roles without relying on predefined hypotheses about what each head might be doing.

Beyond its conceptual contribution, CHG also offers several practical methodological advantages over existing mechanistic interpretability tools. First, because CHG operates directly on next-token prediction, it avoids the need for externally-provided labels [7, 14, 15, 16], controlled input-output pairs [7, 14, 15], or rigid prompt templates [25, 12, 13], which are often required for decoding and interventional approaches. Second, CHG naturally accommodates complex target outputs, including chain-of-thought reasoning [26], where the solution spans multiple intermediate steps. Finally, CHG is highly scalable: it introduces only one learnable parameter per attention head and requires no updates to the underlying model weights, so that the CHG parameters can be fitted in minutes using gradient-based optimization, even for LLMs with billions of parameters. Thus, in settings where analyzing complex dependencies between heads is important, it is feasible to fit large samples of CHG values to estimate a distribution over gating values in a bootstrap fashion.

To test its efficacy, we apply CHG across a diverse set of tasks—mathematical, commonsense, and syntactic reasoning—and across LLMs ranging from 1 to 8 billion parameters with varying training paradigms. We use CHG to analyze not only *where* specific computations take place, but also *how distributed* they are across attention heads, and how these patterns vary across different tasks and models. We also validate the causal scores produced by CHG by comparing them against targeted ablations as well as causal mediation analysis [12, 25], showing strong agreement between predicted and observed effects. Finally, we extend CHG to a contrastive setting to identify distinct sub-circuits that support instruction following versus in-context learning, suggesting that even semantically similar tasks can be underpinned by separable mechanisms.

Our main contributions are fourfold:

1. We introduce causal head gating (CHG), a parametric, scalable method for identifying potentially distributed, task-relevant sub-circuits in transformer models without requiring prompt templates or labeled outputs, and extend it with contrastive CHG to isolate heads supporting specific sub-tasks.
2. We propose a simple causal taxonomy of heads—facilitating, interfering, and irrelevant—that quantifies the effect of each on task performance using CHG-derived scores.
3. We use CHG to show that models contain multiple task-sufficient sub-circuits with varying degrees of overlap, suggesting head roles are not fully modular but depend on interactions with other heads.

4. We use CHG to show that instruction following and in-context learning rely on context-dependent separable circuits at the head level, where CHG-guided gating can selectively suppress one mode without substantially disrupting the other.

The accompanying repository for this paper can be found at https://github.com/andrewnam/causal_head_gating.

2 Related Work

Representational decoders Representational decoders are models trained to map hidden activations to externally labeled properties [7, 14, 15], estimating the mutual information between representations and those properties [27, 28]. However, such probing results are difficult to interpret: simpler decoders may underfit and miss relevant features (false negatives), while complex decoders may overfit and learn spurious correlations (false positives) [29, 30], requiring complexity-accuracy tradeoffs to contextualize results [30]. Moreover, although decodability indicates that a property is encoded in the representation, it does not imply that the model uses that information for its task, highlighting a correlational finding rather than a causal one [31]. Finally, representational decoders require labeled datasets, constraining their use to curated, predefined properties. For a comprehensive review of the probing framework and its limitations, see [27].

Sparse autoencoders (SAE) can be viewed as a related approach, where the autoencoder reconstructs representations through a sparse bottleneck to reveal modular or interpretable features [16, 32]. However, like probing classifiers, their insights remain correlational and still depend on post hoc labeling or interpretation, inheriting the same supervision bottleneck. In contrast, CHG performs direct interventions on model components without external supervision and proposes sufficient sub-circuits to the default unablated model, thereby identifying causal links between attention heads and model behavior on a task.

Causal mediation analysis Causal mediation analysis (CMA) [33, 34] is used to identify the functional roles of specific attention heads by crafting controlled prompt pairs that isolate a hypothesized behavior, then intervening on model components to measure their causal effect on outputs. For instance, in the indirect-object-identification (IOI) task [25], sentences like “When Alice and John went to the store, John gave a drink to...” are used to identify attention heads responsible for resolving coreference. By patching specific head outputs from a source sentence into a structurally matched target, and checking whether the model changes its prediction (e.g. “Alice” instead of “Mary”), CMA localizes the relevant circuit. It has also uncovered head-level roles in function tracking [12], symbol abstraction [13], and other structured settings [35].

However, CMA relies on manually crafted prompt templates and clear mechanistic hypotheses, which limits its scalability to more complex domains. In open-ended tasks like mathematical reasoning [36, 37, 38], the diversity of required knowledge makes it hard to design effective controlled inputs. A single shared template is unlikely to accommodate even two prompts from the MATH dataset [37], such as: “If $\sum_{n=0}^{\infty} \cos^{2n} \theta = 5$, what is $\cos 2\theta$?” and “The equation $x^2 + 2x = i$ has two complex solutions; determine the product of their real parts.” Moreover, LLMs often solve such problems most effectively via chain-of-thought reasoning [26], which unfolds over multiple steps, further complicating the use of a unified prompt structure.

Head ablations Despite the use of multiple heads being commonplace in transformer-based architectures, it has been observed that multiple, and sometimes the majority of, heads can be entirely pruned with minimal impact on model performance [39, 18, 40, 41]. Moreover, entire layers can be pruned while retaining model performance [42, 43, 44]. However, existing works on pruning attention heads have focused primarily on custom-trained small-scale transformers [39, 18, 40] or BERT-based [45] models [41, 43], and the literature is limited for modern causal LLMs such as GPT [46, 1] and Llama [3].

Head pruning has also been used to validate findings from other interpretability methods, such as CMA [25, 13] or attention pattern analysis [18]. In these studies, researchers first identify heads believed to perform specific functions, then ablate them to test their causal impact. Such targeted ablations often lead to disproportionate drops in performance, supporting the hypothesis that those heads are functionally important.

Most closely related to our work are differentiable masking and soft-gating approaches that learn which attention heads to retain or suppress. In [47], the authors apply sparsity gating to identify subcircuits and use the fitted parameters as weighting values in convex combinations for activation patching. Similarly, [48] learns scaling constants for each attention head, but uses the fitted values to identify heads that are most suitable for fine-tuning. Thus, while methodologically similar, our work is unique in applying the gating parameters to identify task-sufficient causal sub-circuits.

Others [18, 40] have opted for hard, binary ablations using the Gumbel-softmax trick [49, 50], fitting gating probabilities rather than weighting parameters. Although these Gumbel-based approaches have been applied for causal circuit discovery in a similar spirit to our work, they suffer from a fundamental limitation that CHG does not. Specifically, while Gumbel-based gating methods also learn differentiable gates per head, they treat each head independently, effectively learning separate Gumbel–Bernoulli distributions for head inclusion. This factorized formulation models only marginal probabilities and cannot capture interdependencies between heads that jointly affect task performance. In contrast, CHG jointly optimizes all gating coefficients under the model’s loss, capturing the full range of interactions and contingencies between the attention heads. Because CHG is highly scalable, it can be fit repeatedly across random seeds or subsets, effectively sampling from the space of sub-circuits without assuming independence between heads. This enables estimation of the underlying distribution over functional head configurations while preserving the joint statistical structure that factorized gating approaches discard.

3 Our Approach: Causal Head Gating

Causal head gating is based on three ideas: applying multiplicative gates to attention heads to evaluate their roles, using regularization to produce variation in the estimates of the gating parameters, and constructing a taxonomy based on that variation. We introduce these ideas in turn.

3.1 Applying gates to attention heads

For a transformer with L layers and H attention heads, we define a gating matrix $G \in [0, 1]^{L \times H}$, where $G_{\ell,h}$ scales the output of head h in layer ℓ , just before the output projection matrix W_ℓ^O (shown in red for an example head in Figure 1a). Given input hidden states $X \in \mathbb{R}^{\text{seq} \times d_{\text{model}}}$, each head computes:

$$A_{\ell,h} = \text{softmax} \left(\frac{XW_Q^{\ell,h}(XW_K^{\ell,h})^\top}{\sqrt{d_k}} \right), \quad V_{\ell,h} = XW_V^{\ell,h}, \quad Z_{\ell,h} = G_{\ell,h} \cdot (A_{\ell,h}V_{\ell,h})$$

where $W_Q^{\ell,h}, W_K^{\ell,h}, W_V^{\ell,h} \in \mathbb{R}^{d_{\text{model}} \times d_k}$ are learned projection matrices for queries, keys, and values.

The gating coefficient $G_{\ell,h}$ modulates the contribution of head h by scaling its output $Z_{\ell,h}$ after attention is applied but before the heads are combined (see Figure 1a). The gated outputs are then

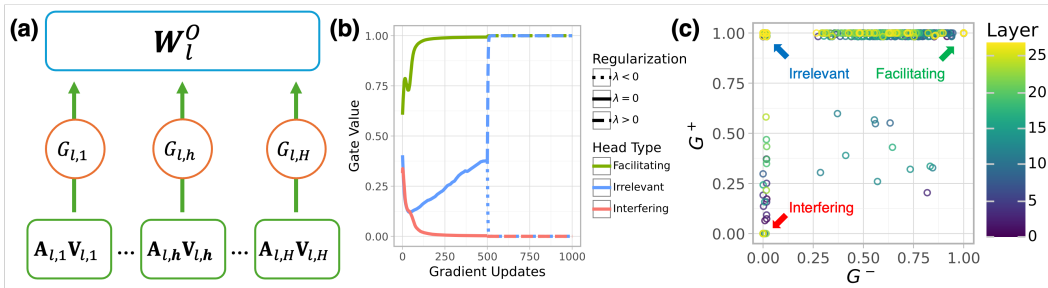


Figure 1: (a) Schematic of a single multihead attention block with CHG-determined gating attenuation (in red). (b) Gate fitting trajectories for three heads on L3.2-3BI with OpenMathInstruct2. When fitting with $\lambda < 0$ and $\lambda > 0$, G^+ and G^- both stay near 1 for facilitating heads and near 0 for interfering heads, but bifurcate to 1 and 0 respectively for irrelevant heads. (c) Gate values after fitting.

Table 1: Causal taxonomy for head roles and corresponding gating patterns.

Role	Description	G^+	G^-	Metric	Ablation Effect
Facilitating	Supports task performance	High	High	G^-	Decreases task performance
Interfering	Interferes with task performance	Low	Low	$1 - G^+$	Increases task performance
Irrelevant	Negligible impact on performance	High	Low	$G^+ \times (1 - G^-)$	No effect on task performance

concatenated and projected:

$$\text{Output}_\ell = \text{Concat}(Z_{\ell,1}, \dots, Z_{\ell,H})W_\ell^O, \quad W_\ell^O \in \mathbb{R}^{Hd_k \times d_{\text{model}}}$$

We fit G by freezing the parameters of the model $\mathcal{M}_\theta$ and minimizing the negative log-likelihood (NLL) on a next-token prediction task with a regularization term specified below.

3.2 Producing variation through regularization

We add a regularization term to the objective that introduces a small but consistent gradient—clipped to ensure NLL remains the dominant term—that nudges the gates for task-irrelevant heads toward 1 or 0 while leaving task-relevant ones relatively unaffected. The NLL optimizes towards improving task performance, and tunes the heads by either increasing the gating values for task-facilitating heads or decreasing the gating values for task-interfering heads. However, if a head does not affect task performance, i.e. is task-irrelevant, then the expected gradient from the NLL is 0, which confounds interpretation of task relevance when evaluating the tuned gating values: a gate $G_{l,h}$ may be close to 1 either because it is important for performing the task (causal), or because gating it has no effect (incidental). We address this limitation by introducing an L_1 -regularization term in our objective function, with weight λ that either nudges gates toward 1 for maximal density ($\lambda > 0$) or toward 0 for maximal sparsity ($\lambda < 0$):

$$\mathcal{L}(G; \mathcal{M}_\theta, \mathcal{D}, \lambda) = - \underbrace{\sum_{(x,y) \in \mathcal{D}} \log P(y \mid x; \mathcal{M}_\theta, G)}_{\text{Negative log-likelihood (NLL)}} - \lambda \underbrace{\sum_{i,j} \sigma^{-1}(G_{i,h})}_{\text{Regularization}} \quad (1)$$

where $\mathcal{M}_\theta$ is the model being analyzed, y is the target text sequence for a given prompt x in dataset $\mathcal{D}$, and σ^{-1} is the clipped inverse-sigmoid function.

We fit G twice: once with $\lambda > 0$ to encourage retention (G^+), and once with $\lambda < 0$ to encourage removal (G^-). To ensure that the heads are aligned across both optimizations, we first fit G with $\lambda = 0$ to establish a shared initialization (see Figure 1), so that any differences between G^+ and G^- reflect only the effect of the regularization and not divergent optimization paths.

3.3 Constructing a taxonomy of task relevance

The G^+ and G^- matrices allow us to interpret the functional role of each head. To formalize this, we introduce a causal taxonomy (Table 1) in which each head is assigned one of three roles—*facilitating*, *interfering*, or *irrelevant*—based on its predicted impact on model performance under ablation. Facilitating heads positively contribute to performance, while ablating them degrades it. Conversely, interfering heads negatively contribute to performance, while ablating them improves it. Finally, irrelevant heads have negligible effect, with ablation leaving performance effectively unchanged.

We instantiate this taxonomy using the fitted CHG matrices G^+ and G^- , which reflect head behavior under opposing regularization pressures. Facilitation is measured by G^- : heads that remain active despite pressure to suppress are likely necessary for the task. Interference is measured by $1 - G^+$: heads that are suppressed even under encouragement to remain are likely harmful. Irrelevance is measured via $G^- \odot (1 - G^+)$, identifying heads that vary in gate values based on regularization.

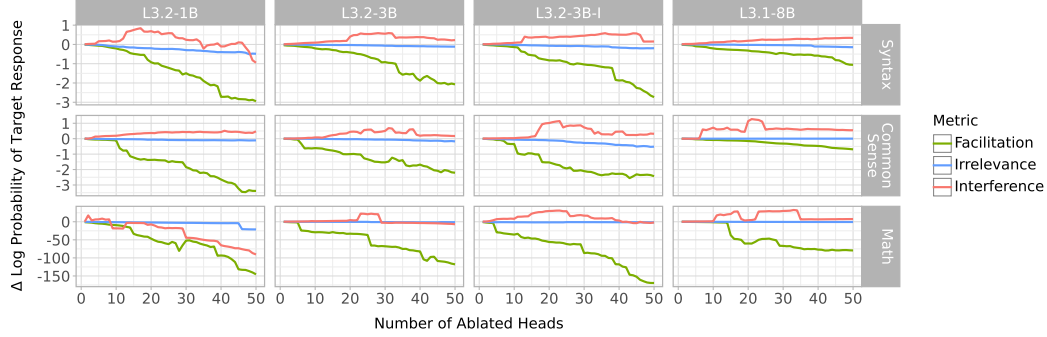


Figure 2: Difference in target log-probability when sequentially setting individual gates in G^+ to 1 and 0 in order of facilitation, irrelevance, and interference scores. The horizontal axis shows the number of heads ablated in descending score order. Positive values indicate task improvement, negative values indicate degradation, and values near zero indicate no effect. Note that not all heads in the top 50 necessarily have high absolute scores.

4 Experiments and analyses

4.1 Causal roles of attention heads

We begin by reporting experiments that evaluated the causal taxonomy presented in Table 1 across four variants of the Llama 3 LLM [3]: L3.1-8B, a pre-trained 8B-parameter model; L3.2-3B, a 3B-parameter model distilled from Llama-3.1-70B (not used in this paper); L3.2-3BI, an instruction-tuned version of Llama-3.2-3B; and L3.2-1B, a 1B-parameter model distilled from L3.1-8B. For each model, we fit CHG matrices on three task types performed over distinct datasets: mathematical reasoning from OpenMathInstruct2 [38], syntactic reasoning from the subset labeled “syntax” in BIG-Bench [51], and commonsense reasoning from CommonsenseQA [52]. We fit CHG matrices independently for each model-dataset pair across 10 random seeds.

We first test whether the causal scores align with the taxonomy’s predictions about performance. Specifically, the taxonomy predicts that, when ablated, attention heads scoring highly on facilitation, irrelevance, or interference should decrease, leave unchanged, or increase the model’s task performance, respectively. To test this, we sort heads in descending order by each causal metric and evaluate the model using the G^+ matrix while toggling each head to 0 or 1 in order of its score. While both G^+ and G^- match the context in which scores were computed, we use G^+ as it retains more heads, providing a more interpretable baseline for ablation. We then compare the retained and ablated masks by the model’s log-probability of the target sequence, expecting the resulting change in log-probability to follow the predicted pattern. As shown in Figure 2, these interventions match the predicted patterns: the difference in target log-probability is negative when progressively ablating facilitating heads, near 0 when ablating irrelevant heads, and positive when ablating interfering heads, up until the set of interfering heads is exhausted.

4.2 Distribution of causal roles

Having validated the causal scores using targeted ablations, we next analyze how they are distributed across models and tasks. Figure 3a shows that for each task, the distribution of head roles is highly consistent across all four model variants. This holds despite large differences in model size (1B to 8B) and training setup (pretraining, distillation, instruction tuning). We quantify these similarities by computing Pearson correlations between head scores across all model pairs for each task and causal metric, yielding 54 model pairs, all of which show high agreement with a minimum correlation of 94.92% and an average of 99.2%. Across tasks, however, we observe notable differences, with the math dataset standing out in particular. For syntax and commonsense reasoning, most heads are irrelevant—63.0% and 64.6% have irrelevance scores ≥ 0.5 , respectively—with only a sparse subset of facilitating heads (25.6% and 27.4% with facilitation scores ≥ 0.5), suggesting that compact, redundant circuits are sufficient for these tasks. In contrast, mathematical reasoning activates a much larger fraction of facilitating heads: 52.6% have facilitation scores ≥ 0.5 , while only 39.0% are

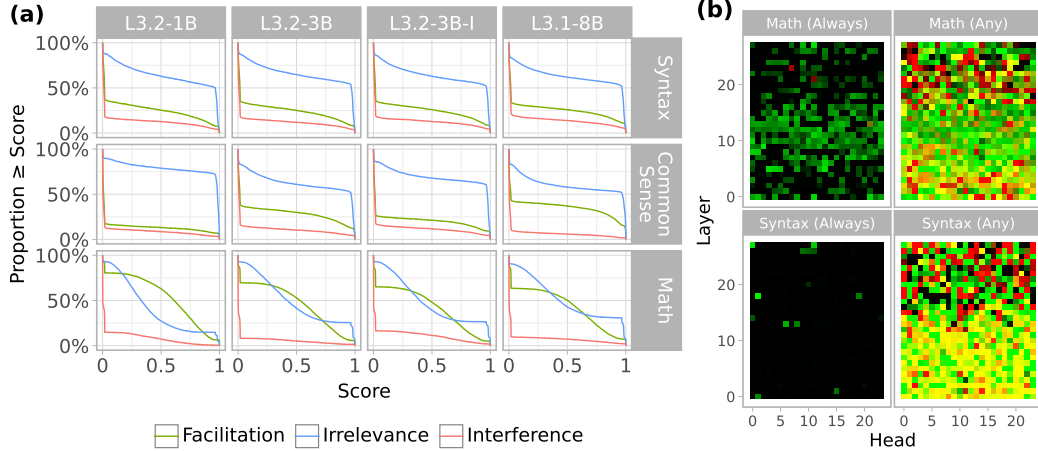


Figure 3: CHG score distributions and consistency. (a) Empirical cumulative distribution of CHG scores across all attention heads, showing the proportion of heads with scores below a given threshold for facilitation, irrelevance, and interference. (b) Aggregated CHG scores on L3.2-3BI, where red and green color channels represent interference ($1 - G^+$) and facilitation (G^-), respectively. Colors are combined using RGB rules: black indicates irrelevance (low in both), and yellow indicates both facilitation and interference (high in both). *Always* aggregates using the minimum across seeds (highlighting consistent effects); *Any* uses the maximum (highlighting any effect across seeds).

Table 2: Percent of heads with facilitation (F) or interference (N) scores ≥ 0.5 across all seeds (always) or in at least one seed (any).

Task	Agg.	L3.2-1B		L3.2-3B		L3.2-3BI		L3.1-8B	
		F	N	F	N	F	N	F	N
Syntax	Always	1.2	0.2	1.5	0.1	0.7	0.0	1.4	0.0
	Any	72.1	57.2	67.9	51.3	72.8	56.1	68.5	59.2
Common Sense	Always	3.9	0.0	4.5	0.0	3.0	0.0	18.7	0.6
	Any	56.6	41.0	75.4	52.4	68.2	55.7	60.3	22.2
Math	Always	38.3	0.4	24.6	1.3	18.3	0.1	25.3	1.0
	Any	81.1	26.0	75.1	13.8	74.4	47.2	75.0	21.2

irrelevant, likely reflecting the task’s higher complexity and need for broader sub-circuitry to support multi-step, latent computations.

It is also worth noting that, across all tasks, 84.0% of heads are marked as facilitating or interfering (score ≥ 0.5) in at least one seed, yet only a small fraction are consistently facilitating or interfering across all seeds (Figure 3b). In syntax and commonsense tasks, most models have fewer than 5% of heads that are always facilitating and virtually none that are always interfering (Table 2). In contrast, math reveals more rigid and consistent circuitry, with up to 38.3% of heads consistently facilitating and 1.3% consistently interfering. These patterns suggest that individual attention heads may not have modular, context-independent roles, but instead participate in a flexible ensemble of overlapping sub-circuits, in which their function depends on the configuration of others [53].

4.3 Comparison with causal mediation analysis

CMA, like CHG, aims to identify attention heads that facilitate task execution, though it does so in a more hypothesis-driven manner. Framed in signal detection terms, CMA and CHG are complementary. CMA exhibits high precision but relatively low sensitivity: while many facilitating heads may go undetected (false negatives), those it does identify are reliably task-relevant (few false positives). Conversely, CHG is biased toward sensitivity over precision. This suggests that heads identified by CMA should also be identified (as showing strong facilitation) under CHG. We test this by comparing

CHG to the results of two former studies using CMA, replicating their methods to identify attention heads with specific computations: heads that encode task information in function vectors [12] and heads that perform symbolic reasoning [13].

For function vectors, we use the six in-context learning tasks used in [12]: ‘antonym’, ‘capitalize’, ‘country-capital’, ‘English-French’, ‘present-past’, and ‘singular-plural’. Each prompt is presented in an in-context learning (ICL) [46] format consisting of 10 input-output examples using a “Q: X\n A: Y” template, followed by a query to be answered. To perform CMA, we corrupt the prompt by randomly shuffling example outputs to induce mismatched pairs, then patch individual head outputs with clean activations to identify which heads recover performance—interpreting high recovery as evidence of causal mediation.

We apply a similar logic to symbolic reasoning tasks from [13], where the goal is to generalize abstract identity rules such as ABA (“flow^Started^flow”) or ABB (“flow^Started^Started”). We deploy the same CMA procedure used in [13] to identify the three-stage symbolic processing mechanism that was reported: (1) *symbol abstraction* heads that abstract symbols (“A” or “B”) away from the actual tokens in the in-context examples; (2) *symbolic induction* heads that operate over the abstracted symbols to induce the symbol for the missing token in the query; (3) *retrieval* heads that retrieve the actual token based on the induced symbol to complete the query. To screen heads of each type, we construct prompt pairs in which either the same token is assigned to different symbols (“A” or “B”) or tokens are swapped while preserving the same rule, and patch activations at certain token positions between them. Attention heads that steer model behavior towards specific hypotheses about the three head types after patching (either converting the abstract rule or altering the actual token) are labeled as mediating. We conduct all experiments on the Llama-3.2-3B-Instruct model.

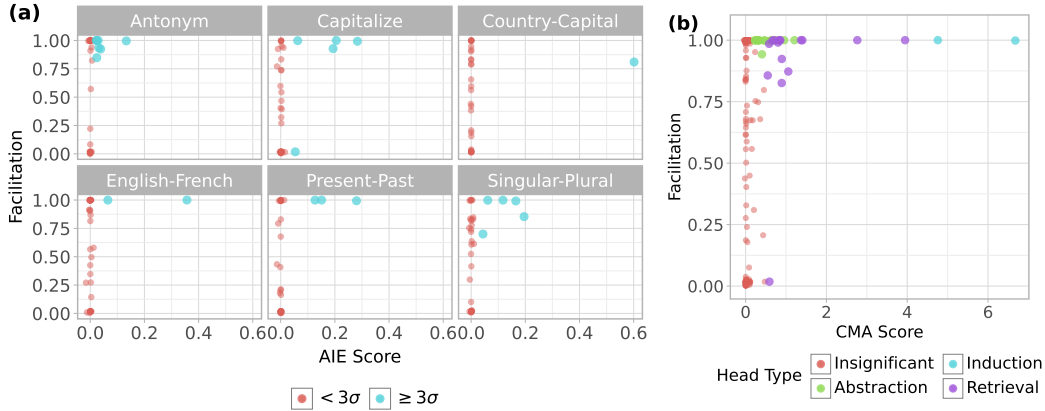


Figure 4: Task-facilitation scores versus (a) average indirect effect for function vector tasks and (b) CMA scores for symbolic reasoning tasks, showing significant heads by type (*abstraction*, *induction*, *retrieval*) and using the maximum CMA score across types for insignificant heads.

As predicted, CMA-identified heads tend to exhibit high facilitation scores under CHG in both domains (Figure 4). To quantify this, we compare the CHG facilitation scores of CMA-identified heads—those with three standard deviations above the mean in function vector tasks or with statistical significance in ABA/ABB tasks [13]—to the remaining ones. Since facilitation and irrelevance depend on the specific sufficient circuit identified by CHG, a head may appear irrelevant in one run but facilitating in another if multiple circuits exist. To account for this, we fit 10 CHG masks per function vector task and 20 per ABA/ABB task, and compute each head’s maximum facilitation score across runs—capturing whether it participates in any sufficient circuit. We find significantly greater facilitation among mediating heads in both the function vector tasks ($t(23.05) = 8.52, p < 10^{-8}$) and the ABA/ABB tasks ($t(53.77) = 11.18, p < 10^{-15}$), supporting the relationship between CMA and CHG-identified task relevance.

4.4 Contrastive Causal Head Gating

The results above indicate that CHG effectively distinguishes among facilitating, irrelevant, and interfering attention heads. However, as an exploratory method, it lacks the granularity to charac-

terize the specific functions of these subnetworks. For instance, consider the ‘antonym’ task from Section 4.3, presented in an in-context learning (ICL) format with 10 examples and a single-word response, as defined in [12]. To perform this task successfully, the model must not only generate the appropriate antonym of a given word, but also infer the task itself from the 10 input-output pairs in the prompt. Thus, a minimal circuit of task-facilitating heads will contain both those involved in task inference and those involved in antonym production, and CHG cannot distinguish between the two. This becomes more pronounced as task complexity increases, as in the OpenMathInstruct2 dataset, where the minimal circuit must jointly support diverse sub-tasks, including English comprehension, mathematical reasoning, chain-of-thought processing, and LaTeX generation.

To address this, we introduce a simple extension of CHG that not only identifies facilitating heads for a given task but also isolates the sub-circuit responsible for a particular sub-task. We generate parallel variants of the same task that share all features except for a controlled difference in the required operation, allowing us to isolate the corresponding sub-circuits. In doing so, we take a step toward a hypothesis-driven approach, decomposing the task into sub-steps while remaining agnostic to the mechanistic implementations. For example, the antonym task can be constructed as an ICL task using the default format from [12], or as an instruction-following task where the model is presented with the task description “Given an input word, generate the word with opposite meaning”. By comparing the resulting attention circuits, we can disentangle components responsible for task inference from those involved in antonym generation.

Furthermore, rather than simply applying CHG to each version and directly comparing the results, we propose a combined approach that fits a single mask with a joint objective to forget one variant of the task while retaining the other, so that the resulting gate matrix suppresses heads uniquely necessary for one variant but dispensable for the other:

$$\mathcal{L}(G; \mathcal{M}_\theta, \lambda) = \sum_{(x_R, y_R, x_F, y_F)} \log P(y_F | x_F) - \log P(y_R | x_R) - \lambda \sum_{i,j} \sigma^{-1}(G_{i,h}) \quad (2)$$

where $\log P(y | x)$ denotes the log-probability of target sequence y given prompt x under model $\mathcal{M}_\theta$ with gating matrix G , the sum ranges over matched tuples (x_R, y_R, x_F, y_F) of the retention and forget variants that differ only in task formulation, and $\lambda > 0$. To stabilize the gradient, we clip the inverse-sigmoid as in Eq. 1 as well as the difference in log-probability.

We evaluate this method using the six function vector tasks from Section 4.3, leveraging the natural language task descriptions provided in [12] to construct instruction-based variants. For each problem, we replace the 10-shot word-pair examples with a prompt containing the task instruction and a single example. We then fit the *contrastive* causal head gating (CCHG) mask to forget the ICL variant of five tasks while retaining the instruction-based format, holding out the sixth task for evaluation. If task inference from examples, instruction-following, and task execution are indeed mediated by separable circuits, this analysis should disable example-based generalization while preserving instruction-based performance. We perform our experiments in both directions (forgetting ICL while retaining instruction-following, and vice versa), using each of the six tasks as the held-out evaluation task. All experiments were conducted on the LLaMA-3.2-3B-Instruct model.

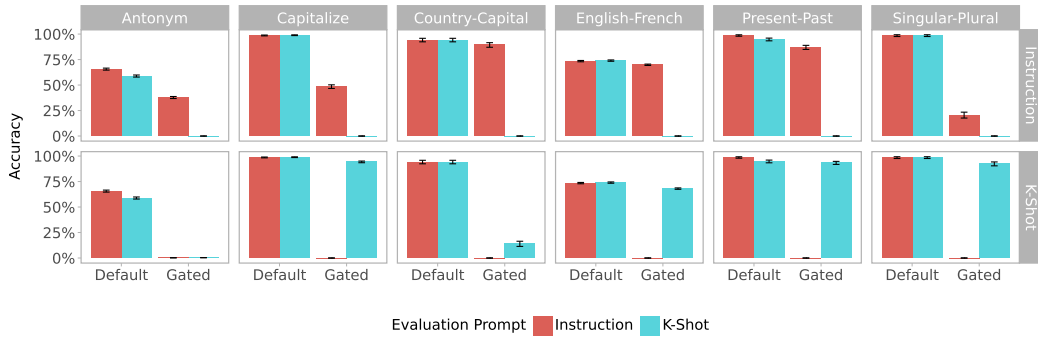


Figure 5: Task accuracy under CCHG. Columns indicate held-out evaluation tasks and rows indicate the retained prompt format. Bar color shows the evaluation prompt format. “Default” and “gated” indicate whether CCHG is applied during evaluation. Error bars indicate 95% CI.

As shown in Figure 5, the CCHG masks generalize to the held-out task. When the model is induced to forget task inference from ICL examples across five tasks, its target task accuracy drops to zero on the ICL variant of the held-out task while in most cases remaining well above zero—and often close to the unablated baseline—on the instruction-based variant. A similar pattern emerges when forgetting is applied using the instruction-based format: performance collapses on instruction prompts while generally remaining intact for example-based ones.

Interestingly, while degradation is often small for the retained prompt format, this pattern is not consistent across all tasks. For example, when the gating matrix is fitted to retain ICL and forget instruction-following, the ‘singular-plural’ task shows only a small drop in ICL accuracy (98% $\rightarrow$ 92%) but a complete failure on instruction prompts (98% $\rightarrow$ 0%). When this setup is reversed—fitted to retain instruction-following and forget ICL—accuracy on ICL drops from 98% to 0%, while instruction accuracy drops more modestly (98% $\rightarrow$ 21%). Across the 6 tasks, 3 (‘country-capital’, ‘English-French’, ‘present-past’) remain robust as held-out tasks under instruction prompts, and 4 (‘capitalize’, ‘English-French’, ‘present-past’, ‘singular-plural’) do so under ICL prompts.

Thus, our results indicate that the circuits for instruction following and ICL may be separable at the head level. However, this separability also depends on the task, suggesting that task execution circuits may share heads with those used for task understanding and representation.

5 Discussion

In this work, we introduced Causal Head Gating (CHG), a flexible and scalable method for identifying causally relevant attention heads in large language models. CHG assigns each head a graded score for facilitation, interference, or irrelevance based on its effect on task performance, going beyond correlational or observational analyses. These scores predict performance changes under targeted ablations, confirming that facilitation, interference, and irrelevance scores capture causal impact. Crucially, it does so using next-token prediction alone, thereby avoiding reliance on labeled data or handcrafted prompts, making it broadly and easily applicable. Moreover, CHG requires no finetuning or auxiliary decoder model, and introduces only one parameter per head, allowing it to run in minutes even on billion-scale models. To validate our method, we demonstrated that existing works within the mechanistic interpretability literature successfully corroborate our findings using, and that the ICL and instruction-following circuits revealed using contrastive CHG successfully generalize across tasks.

Interestingly, across the range of models and tasks we investigated, we observed that attention heads form task-sufficient sub-circuits with low overlap. Moreover, a single head may vary in its relevance across multiple runs depending on which others are active, reflecting the distributed and context-dependent nature of computation in LLMs, and in rare cases, a head may even receive low G^+ but high G^- scores within the same run. We hypothesize that this variability reflects an interaction-dependent landscape in which causal roles shift with circuit configuration. While these complexities may appear messy, we view them as a strength of CHG, revealing the redundancy and interdependence that underlie emergent model behavior. Because CHG is highly scalable, it can be repeatedly applied to estimate distributions over gating values, providing a bootstrapped view of redundant and contingent sub-circuits with greater fidelity to the model’s underlying dependency structure.

While CHG provides a lightweight and scalable approach for exploratory analysis, requiring only a dataset and no model finetuning or supervision, it is not designed to reveal the precise computations performed by individual heads. Instead, CHG offers a complementary first-pass diagnostic tool that identifies candidate heads or sub-circuits with consistent causal influence, guiding where more granular, hypothesis-driven methods such as causal mediation or activation patching can be applied. In this way, CHG provides a practical entry point into large-scale causal interpretability, mapping functional dependencies that subsequent analyses can examine in greater detail.

We hope that our work encourages further exploration of causal structure in language models as a foundation for more mechanistic understanding. Future work may build on these tools to develop circuit-level explanations of how models implement complex behaviors.

Acknowledgments and Disclosure of Funding

We thank Declan Campbell and Alexander Ku for helpful discussions, and Legasse Remon for assistance with dataset organization.

Jonathan Cohen was supported by the Vannevar Bush Faculty Fellowship, sponsored by the Office of Naval Research.

The authors declare no competing interests.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [2] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.
- [3] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- [4] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.
- [5] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does LLM safety training fail? *Advances in Neural Information Processing Systems*, 36:80079–80110, 2023.
- [6] Laura Weidinger, Joslyn Barnhart, Jenny Brennan, Christina Butterfield, Susie Young, Will Hawkins, Lisa Anne Hendricks, Ramona Comanescu, Oscar Chang, Mikel Rodriguez, et al. Holistic safety and responsibility evaluations of advanced ai models. *arXiv preprint arXiv:2404.14068*, 2024.
- [7] Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. *arXiv preprint arXiv:1905.05950*, 2019.
- [8] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermy, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2, 2023.
- [9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.
- [10] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivi re, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.
- [11] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. <https://transformer-circuits.pub/2021/framework/index.html>.
- [12] Eric Todd, Millicent L Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. Function vectors in large language models. *arXiv preprint arXiv:2310.15213*, 2023.
- [13] Yukang Yang, Declan Campbell, Kaixuan Huang, Mengdi Wang, Jonathan Cohen, and Taylor Webb. Emergent symbolic mechanisms support abstract reasoning in large language models. *arXiv preprint arXiv:2502.20332*, 2025.
- [14] Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. What do you learn from context? Probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*, 2019.
- [15] John Hewitt and Christopher D Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, 2019.

- [16] Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermy, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. *Transformer Circuits Thread*, 2024.
- [17] Andrew Lee, Xiaoyan Bai, Itamar Pres, Martin Wattenberg, Jonathan K Kummerfeld, and Rada Mihalcea. A mechanistic understanding of alignment algorithms: A case study on dpo and toxicity. *arXiv preprint arXiv:2401.01967*, 2024.
- [18] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*, 2019.
- [19] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.
- [20] Geoffrey E Hinton. Learning distributed representations of concepts. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 8, 1986.
- [21] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/toy_model/index.html.
- [22] Kayson Fakhar and Claus C Hilgetag. Systematic perturbation of an artificial neural network: A step towards quantifying causal contributions in the brain. *PLOS Computational Biology*, 18(6):e1010250, 2022.
- [23] Tyler Giallanza, Declan Campbell, Jonathan D Cohen, and Timothy T Rogers. An integrated model of semantics and control. *Psychological Review*, 2024.
- [24] Melanie Mitchell. *Complexity: A guided tour*. Oxford University Press, 2009.
- [25] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. *arXiv preprint arXiv:2211.00593*, 2022.
- [26] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.
- [27] Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.
- [28] Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. Information-theoretic probing for linguistic structure. *arXiv preprint arXiv:2004.03061*, 2020.
- [29] John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. *arXiv preprint arXiv:1909.03368*, 2019.
- [30] Elena Voita and Ivan Titov. Information-theoretic probing with minimum description length. *arXiv preprint arXiv:2003.12298*, 2020.
- [31] Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. Probing the probing paradigm: Does probing accuracy entail task relevance? *arXiv preprint arXiv:2005.00719*, 2020.
- [32] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

- [33] Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. *Advances in Neural Information Processing Systems*, 33:12388–12401, 2020.
- [34] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.
- [35] Zifan Zheng, Yezhaohui Wang, Yuxin Huang, Shichao Song, Mingchuan Yang, Bo Tang, Feiyu Xiong, and Zhiyu Li. Attention heads of large language models. *Patterns*.
- [36] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- [37] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.
- [38] Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. Openmathinstruct-2: Accelerating ai for math with massive open-source instruction data. *arXiv preprint arXiv:2410.01560*, 2024.
- [39] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Advances in Neural Information Processing Systems*, 32, 2019.
- [40] Jiaoda Li, Ryan Cotterell, and Mrinmaya Sachan. Differentiable subset pruning of transformer heads. *Transactions of the Association for Computational Linguistics*, 9:1442–1459, 2021.
- [41] Mengzhou Xia, Zexuan Zhong, and Danqi Chen. Structured pruning learns compact and accurate models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1513–1528, 2022.
- [42] Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. *arXiv preprint arXiv:1909.11556*, 2019.
- [43] Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. On the effect of dropping layers of pre-trained transformer models. *Computer Speech & Language*, 77:101429, 2023.
- [44] Shwai He, Guoheng Sun, Zheyu Shen, and Ang Li. What matters in transformers? not all attention is needed. *arXiv preprint arXiv:2406.15786*, 2024.
- [45] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.
- [46] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.
- [47] Nicola De Cao, Michael Schlichtkrull, Wilker Aziz, and Ivan Titov. How do decisions emerge across layers in neural models? interpretation with differentiable masking. *arXiv preprint arXiv:2004.14992*, 2020.
- [48] Fangcong Yin, Xi Ye, and Greg Durrett. Lofit: Localized fine-tuning on llm representations. *Advances in Neural Information Processing Systems*, 37:9474–9506, 2024.
- [49] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.
- [50] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

- [51] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.
- [52] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.
- [53] Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. Talking heads: Understanding inter-layer communication in transformer language models. *Advances in Neural Information Processing Systems*, 37:61372–61418, 2024.
- [54] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

6 Technical Appendices and Supplementary Material

6.1 Datasets

For each dataset, we split the full set into three partitions: an example set, a training set, and a validation set. Example problems were selected from the top K shortest prompt-solution sequences after tokenization. One example was randomly drawn from the example set to be included in each training/validation prompt to help align model responses with the task format. For multiple-choice datasets, answer options were randomly shuffled and labeled with capital letters (A, B, C, ...), and the target answer was the correct letter.

OpenMathInstruct2 We used the `OpenMathInstruct-2_train_1M` subset. We filtered out problems marked as having no solution, removed duplicate prompts (even if their solutions differed), and retained the 55,050 shortest prompt-solution pairs by total tokenized length. From this, we selected 50 examples, 50,000 training problems, and 5,000 validation problems. Each prompt began with the instruction: “For each problem, explain your reasoning step by step and use LaTeX for all mathematical expressions. Indicate your final answer using `\boxed{...}`.”

CommonsenseQA We selected 10 problems for the example set, then split the remaining data into a 90% / 10% training/validation split.

BIG-Bench syntax We included all tasks labeled ‘syntax’ in BIG-Bench: ‘linguistic mappings’, ‘tense’, and ‘subject-verb-agreement’. The ‘linguistic mappings’ category consisted of five subtasks, each with its own instruction:

- **Past tense:** “Convert the verb to its past tense form.”
- **Plural:** “Convert the noun to its plural form.”
- **Pronoun replacement:** “Replace the repeated name with the correct pronoun.”
- **Question formation:** “Convert the statement into a yes/no question.”
- **Sentence negation:** “Convert the statement into a negative sentence.”

The ‘tense’ task used the instruction: “Modify the tense of a given sentence.”

The ‘subject-verb-agreement’ task used the instruction: “Choose the grammatically correct verb form that agrees with the subject of the sentence.”

Each task or subtask was treated independently for splitting and prompt generation. We allocated 10 examples per subtask, with a 90% / 10% split over the remainder into training and validation. Example problems used in prompts were always drawn from the same subtask as the target problem.

Function vector tasks We included six tasks: ‘antonym’, ‘capitalize’, ‘country-capital’, ‘english-french’, ‘present-past’, and ‘singular-plural’. Each task was used in two formats: 10-shot in-context learning (ICL) prompts with 10 input-output pairs, and instruction-based prompts using task descriptions from [12]:

- **Antonym:** “Given an input word, generate the word with opposite meaning.”
- **Capitalize:** “Given an input word, generate the same word with a capital first letter.”
- **Country-Capital:** “Given a country name, generate the capital city.”
- **English-French:** “Given an English word, generate the French translation of the word.”
- **Present-Past:** “Given a verb in the present tense, generate the verb’s simple past inflection.”
- **Singular-Plural:** “Given a singular noun, generate its plural inflection.”

We allocated 10 examples per task, and split the remaining data into 90% training and 10% validation. Example problems used in prompts matched the format (ICL or instruction) of the task being evaluated.

Symbolic reasoning (ABA/ABB) : We procedurally generated symbolic reasoning prompts following the A^*B^*A and A^*B^*B templates from [13]. using 4 in-context examples per prompt. Each prompt was generated by selecting 10 random tokens—8 assigned to the 4 examples and 2 used in the query. We used individual tokens rather than full words, since multi-token words often behave similarly: once the first token is generated, the model tends to complete the rest automatically, reducing the task to token-level pattern recognition.

6.2 Training details

Causal head gating For each model and task, we first fit a CHG matrix G with $\lambda = 0$ for 500 gradient updates with a batch size of 64 samples. G was initialized with random values sampled uniformly between 0 and 1. We used the Adam optimizer [54] for optimization using an initial learning rate of 0.1 with a linear decay that terminates with a learning rate of 0.01. After fitting G with $\lambda = 0$, we fit G^+ and G^- using G as the initial conditions and $\lambda = \pm 0.1$ for 500 gradient updates with an initial learning rate of 0.5 and a terminal learning rate of 0.1. We clipped the regularization term at ± 4 .

Contrastive causal head gating For each model and task pair, we fit a CCHG matrix G with $\lambda = -0.1$, clipping the regularization term at 4 and the log-probability difference at 5. We fitted G over 500 gradient updates with a batch size of 64 using the Adam optimizer with an initial learning rate of 0.1 with a linear decay that terminates with a learning rate of 0.01.

6.3 Hardware and compute

For all our experiments, we used 128 GB of CPU RAM and a single Nvidia H100 GPU at a time. Each run of CHG (1,500 gradient updates) took between 15 minutes and 1 hour, depending on the model and dataset. Each run of CCHG (500 gradient updates) took approximately 5 minutes. We estimate that all experiments reported in this paper can be completed in under 100 GPU hours. Preliminary or failed experiments required negligible additional compute and are not included in the total.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: We claim in the abstract and introduction that CHG offers causal insight into individual attention heads in LLMs, which we substantiate using ablation analysis and comparison to CMA. We also claim that instruction following and ICL are separable at the head level, which we show using CCHG. Lastly, we claim that LLMs contain multiple sparse sub-circuits that are individually sufficient for different tasks, which we show in our consistency analysis.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: The paper discusses key limitations in the Discussion section, including CHG's inability to explain why heads matter, occasional divergence between G^+ and G^- , and the context-dependent variability in head roles across runs.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: Our paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide detailed descriptions of the CHG algorithm, datasets, model variants, and evaluation methods (e.g., ablation, CMA comparisons). The precise methods for reproducing the CMA results are better described in the original papers. Hyperparameters and additional procedural details are included in the supplementary materials.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The accompanying repository can be found at https://github.com/andrewnam/causal_head_gating. We also note that the models and datasets used in our paper are publicly available.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The training and evaluation details, hyperparameters, and optimizer settings can be found in the supplementary materials.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The paper reports t-test results for CMA comparisons and includes 95% confidence intervals in the CCHG evaluation plots

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.

- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
 - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.
8. **Experiments compute resources**
 Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?
 Answer: [Yes]
 Justification: Details on compute resources are provided in the supplementary materials.
 Guidelines:
- The answer NA means that the paper does not include experiments.
 - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
 - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
 - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).
9. **Code of ethics**
 Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?
 Answer: [Yes]
 Justification: We have reviewed the NeurIPS Code of Ethics and confirm that all aspects of our research fully comply.
 Guidelines:
- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
 - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
 - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).
10. **Broader impacts**
 Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?
 Answer: [NA]
 Justification: The paper presents foundational interpretability methods without direct application or deployment, and we do not foresee societal impacts resulting from its current scope.
 Guidelines:
- The answer NA means that there is no societal impact of the work performed.
 - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
 - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
 - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
 - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks, as it uses only publicly available models and datasets and does not release any new high-risk assets.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All assets are cited in the paper.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: We do not release any new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: Our paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Our paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Our study is applied directly to LLMs and is central to our methodology.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.