# WHAT'S THE PLAN? METRICS FOR IMPLICIT PLANNING IN LLMS AND THEIR APPLICATION TO RHYME GENERATION

### **Anonymous authors**

Paper under double-blind review

#### ABSTRACT

Prior work suggests that language models, while trained on next token prediction, show implicit planning behavior: they may select the next token in preparation to a predicted future token, such as a likely rhyming word, as supported by a prior qualitative study of Claude 3.5 Haiku using a cross-layer transcoder. We propose much simpler techniques for assessing implicit planning in language models. With the focus on the case study on rhymed poetry generation, we demonstrate that our methodology easily scales to many models. Across models, we find that the generated rhyme can be manipulated by steering at the end of the preceding line with a vector representing e.g. a "rhyme with -ight" feature, affecting the generation of intermediate tokens leading up to the rhyme. We show that implicit planning for rhyme families is a universal mechanism, present in smaller models than previously thought, starting from 1B parameters. This shows that the phenomenon of rhyming offers a widely applicable direct way to study implicit planning abilities of LLMs. More broadly, understanding planning abilities of language models can inform decisions in AI safety and control.

# 1 Introduction

Does the remarkable ability of modern language models in generating coherent text result from some form of implicit planning? Our paper develops methods for investigating this question, and applies them to the case study of rhymed poetry.

In language production, humans are known to use various strategies to plan ahead for words that they utter (Ferreira & Swets, 2002; Lee et al., 2013; Huettig, 2015; Barthel et al., 2016). Similar mechanisms could plausibly apply in Transformer language models. In the context of text generation with language models, *planning* for goal token(s) is a pattern that implicates several aspects.

**Definition 1.1. Forward planning** is the creation of planning representations, which encode properties of goal token(s) at later steps.

**Definition 1.2.** Forward planning is **successful** if it creates representations that causally implicate the generation of goal tokens.

**Definition 1.3.** Planning is **explicit** if planning representations are model outputs, and **implicit** if planning representations are part of hidden activations.

**Definition 1.4. Backward planning** is relying on planning representations for generating intermediate tokens before the goal token(s) are produced.

**Definition 1.5.** Backward planning is **successful** if generated intermediate tokens increase the likelihood of generating goal token(s).

By its nature, *implicit* planning is harder to study than explicit planning, e.g. laying out the plan for a task in a chain of thought. To establish implicit planning behavior, we need to show that forward planning is successful and accompanied with backward planning, which, ideally, is also successful. In this study, we focus on rhyming as a phenomenon where planning ahead for the kinds of words to produce is particularly transparent, as suggested by the pioneering case study in Lindsey et al. (2025).

Rhymed poetry generation offers a unique window into planning, because it is linked to text elements (rhyming words) whose nature and position are predictable from general principles but not determined by immediately preceding tokens.

The contributions of our work include:

- A rhyming dataset of 1050 lines from 10 rhyme families.
- A set of quantitative metrics for assessing aspects of rhyme planning behavior: successful forward planning, backward planning, and successful backward planning.
- Evaluation of a set of 23 diverse open weight language models (from 1B to 32B parameters).

Our study focuses on rhymed poetry and uses localized steering intervention. We show that mean activation difference steering works across the board, altering forward and backward planning. We find a lot of similarities, but also some differences between models we study (cf. Appendix E and F).

The rest of the paper is organized as follows. Section 2 provides the scientific context; section 3 explains the details of our methodology and the experimental setup. Section 4 reports the results of our experiments, comparing language models according to our metrics. In section 5, we discuss some observations on the planning circuits, which parallel Lindsey et al.'s findings for Haiku. We summarize our work and discuss its limitations and further directions in section 6.

## 2 RELATED WORK

One piece of evidence for implicit planning in language models comes from the studies of unreliable chain of thought (CoT): "When we bias models toward incorrect answers, they frequently generate CoT explanations rationalizing those answers" Turpin et al. (2023); this also applies to newer reasoning models (Chen et al., 2025). When rationalizing, the model must be entertaining the suggested answer before the rationalizing CoT is generated. So a critical output can be implicitly planned long before it is produced, an instance of implicit *forward* planning. Furthermore, intermediate steps leading up to the answer (in this case, the chain of thought) are conditioned on the implicit plan, an instance of *backward* planning.

As a response to reports on LLMs producing useful plans in domains such as coding (Bairi et al., 2023, e.g.), skeptical arguments about the general planning ability of LLMs have been raised in the literature. Various studies (Valmeekam et al., 2023; Zhang et al., 2024; Stein et al., 2025) show empirically that explicit planning abilities of current LLMs are limited. The conceptual argument that "a system that takes constant time to produce the next token cannot possibly be doing principled reasoning on its own" Kambhampati et al. (2024) naturally applies to implicit planning as well.

However, LLMs can plausibly engage in some forms of simpler, heuristic type of planning – just like humans, whose general planning capacities are limited (Kahneman & Tversky, 1977; Buehler et al., 2010), are known to still plan ahead in their speech production (Brown-Schmidt & Konopka, 2015). Indeed, shaping the rhythm and rhyme for the next poetic line is a task that can be solved well enough with limited planning capacities. This is supported by the success of previous poetry generation experiments that used relatively simple tools Hopkins & Kiela (2017); Lau et al. (2018); Ormazabal et al. (2022); Jhamtani et al. (2019); Ghazvininejad et al. (2016).

Implicit planning representations in LLMs have been found for specific models (Pochinkov et al., 2024; Men et al., 2024; Wu et al., 2024). In this paper, we aim to compare planning across a wider range of language models. We zoom in rhyming poerte generation, inspired by Lindsey et al. (2025). The authors gave Claude Haiku 3.5 the prompt:

Example 1. A rhyming couplet:\n He saw a carrot and had to grab it\n

The model was able to complete the second line with the correct rhyme, e.g. *His hunger was like a starving rabbit*. Furthermore, elements of *implicit forward planning* mechanisms were identified: there are vector components in the activations of the token at the end of the first rhyming line (second \n above) that Lindsey et al. identify as representing planning for the rhyme of the next line. Some of these planning activation components, they argue, correspond to the potential rhyme *rabbit* and, if

suppressed, lead to other rhyming words (such as *habit*) being generated much more often. It is even possible to inject a planning direction for a word such as *green* that does not fit the context, causing the model to produce it instead of a correct rhyme.

In addition, Lindsey et al. found evidence of *backward planning* in the rhyming context. The planning directions, they argue, influence the intermediate words. If *rabbit* is the dominant planned rhyme word, certain constructions are produced, e.g. comparison construction as in *His hunger was like a starving rabbit*. If *rabbit* planning directions are artificially suppressed, different intermediate parts of the line are generated, such as *His hunger was a powerful habit*.

Lindsey et al.'s evidence, while persuasive, is limited to Claude Haiku's behavior on a small number of examples. Our paper addresses limitations of their study. First, we assess forward and backward planning in rhyming across a range of models. Second, we introduce several quantitative metrics based on a varied dataset of rhymes. Third, we address the complexity problem: the cross-layer transcoder (CLT) approach used by Lindsey et al., while offering diverse interpretability promises, is particularly complex, computationally expensive, and difficult to replicate. Just training a CLT for a single model of modest size is estimated to require days of compute on a highly performant and expensive GPU such as H100 (Ameisen et al., 2025, Appendix D).

## 3 METHODOLOGY

Taking inspiration in the observations of Lindsey et al., we propose several metrics for quantifying implicit forward and backward planning, and apply them to a variety of language models in the context of rhymed text generation. We propose simpler methods that do not involve the costly training and use of cross-layer transcoders, or other types of dictionary learning. We focus here on planning for a rhyme family, which can be manipulated robustly; see Appendix E for a discussion of model planning to produce specific rhyming words.

#### 3.1 Dataset Creation

We call the set of words that all rhyme with each other a rhyme family. Words in a rhyme family tend to share a suffix, which we use to name the rhyme family. For example, the -ing rhyme family contains words such as king and ring. We chose a set of 10 rhyme families (-ing, -air, -ip, -oat, -ird, -ee, -ight, -ake, -ow, -it). We also chose 20 pairs of rhyme families such that every rhyme family is the first in two pairs and the last in two pairs  $^1$ . For each rhyme family (RF), we generate 105 rhyming couplets using Claude 3.5 Sonnet, regenerating when the output did not actually match the rhyme family. from each generated couplet, we remove the second line and concatenate the string "A rhyming couplet:\n" before the first line. We split the generated lines at random into a train and test set with 85 of the 105 lines in the train set  $\mathbf{P}_{\mathbf{RF}}^{\mathbf{Train}}$  and 20 in the test set  $\mathbf{P}_{\mathbf{RF}}^{\mathbf{Test}}$ .

#### 3.2 Models

We test rhyme planning in both base and instruction tuned models from four language model families (Gemma2, Gemma3, Qwen3 and Llama3.1/3.2) with model sizes between 1B and 32B parameters Team et al. (2024; 2025); Yang et al. (2025); Grattafiori et al. (2024). For every model we test both the base and the instruction tuned versions. <sup>2</sup>

#### 3.3 MEAN ACTIVATION STEERING AND CHOOSING A STEERING VECTOR

If a model exhibits implicit forward planning for a rhyme, then an intervention on an early position (such as the newline in 1) can alter the rhyme produced.

We estimate a steering vector using the average activation difference like in Arditi et al. (2024). For example, we calculate the average activation of the newline ( $\setminus$ n) token after lines that rhyme with *sick*, and subtract it from the average activation of the newline ( $\setminus$ n) token after lines that rhyme with *pain*. The resulting steering vector, when applied to the newline token after the line *The house was* 

<sup>&</sup>lt;sup>1</sup>The exact rhyme family pairs used can be found in Appendix A.

<sup>&</sup>lt;sup>2</sup>Only Qwen3 32B did not have an open weight base model available.

built with sturdy, reddish brick, can lead to 'rhymed' generations like And stood for years, enduring wind and rain instead of the unsteered version And stood for years, enduring every trick.

For each model, layer l and rhyme family pair  $(\mathbf{RF_1}, \mathbf{RF_2})$  we extract a steering vector  $s_{\mathbf{RF_1} \to \mathbf{RF_2}}^{(l)}$  by calculating the mean activation difference between the activations on their train sets

$$s_{\mathbf{RF_1} \to \mathbf{RF_2}}^{(l)} = m \cdot \left( \sum_{p \in \mathbf{P_{RF_1}^{Train}}} \mathbf{x}_{\mathbf{pos}(\backslash \mathbf{n}) - 1}^{(l)}(p) - \sum_{p \in \mathbf{P_{RF_1}^{Test}}} \mathbf{x}_{\mathbf{pos}(\backslash \mathbf{n}) - 1}^{(l)}(p) \right)$$

Here m is a constant, in our case set to  $1.5,^3 \mathbf{x}_k^{(l)}$  is the hidden activation of the model at layer l and position k,  $\mathbf{pos}(\mathbf{n})$  is the position of the newline token in p and  $\mathbf{pos}(\mathbf{n}) - 1$  is the token position before the newline token (the last token of the first line, most often containing a word from the rhyme family).

To apply a steering vector, we add it to the residual stream  $\mathbf{x}_{\mathbf{k}}^{(1)}$  on the correct layer and token position during generation. Notice that **we only apply the steering vector on one token only** (the last token of the first line or the newline token).

For each steering vector we measure the average Fraction of Correct Rhyme Family (Steered) metric described below on  $\mathbf{P_{RF_2}^{Test}}$  with 50 samples for each of the 20 prompts. For the final steering vector of each model and rhyme family  $s_{\mathbf{RF_1} \to \mathbf{RF_2}}$  we choose out of all layers, the layer whose steering vector had the maximum score in this metric.

While we opt for average activation difference steering because of its simplicity, it is also possible to obtain a steering vector with other methods, such as differences of classifier probe weights or SAE weights. For examples of generated outputs with alternative steering vectors, see Appendix G.

#### 3.4 METRICS

Let  $C_{RF}$  be a collection of 1000 couplets generated by some model using the prompts in  $P_{RF}^{Test}$ . We sample 50 generation per prompt  $p \in P_{RF}^{Test}$ .

Let  $C_{RF_1 \to RF_2}$  be a collection of 1000 couplets generated by some model using the prompts in  $P_{RF}^{Test}$  while being steered with  $s_{RF_1 \to RF_2}$ .

Let  $Y_{RF}$  be a collection of sequences of probability distributions over text tokens generated by using  $C_{RF}$  as input to a model.

Let  $Y_{\mathbf{RF_1} \to \mathbf{RF_2}}$  be a collection of sequences of probability distributions over text tokens generated by using  $C_{\mathbf{RF_1}}$  as input to a model while steering with  $s_{\mathbf{RF_1} \to \mathbf{RF_2}}$ .

As evidence *successful forward planning*, we use effectiveness of the steering intervention: if at position X there was no planning representation for a later position Y, then intervening on position X would not have changed the outcomes for position Y in a predictable way. This is done by comparing the following two metrics.

Fraction of Correct Rhyme Family. We calculate the fraction of couplets in  $C_{\mathbf{RF}}$ , where the last word has the correct rhyme family (rhymes with the last word of the first). We made collections containing all words in each rhyme family to do this.

Fraction of Correct Rhyme Family (Steered). This metric is calculated the same way as the Fraction of Correct Rhyme Family metric. We evaluate it separately for each rhyme family pair  $(\mathbf{RF_1},\mathbf{RF_2})$  on  $\mathbf{C_{RF_1\to RF_2}}$ . We assess **backward planning** with the help of probability based metrics. Successful steering means that planning representations can be manipulated successfully. This allows us to assess *backward planning* by measuring to what extent interventions on the planning activations affect the model's behaviors at intermediate positions. For example, Gemma2 9B model

<sup>&</sup>lt;sup>3</sup>Mean activation difference estimation defines an inherent scale: steering vector can be used with multiplier of 1. While that often works, we found that slightly bigger values such as 1.5 or 2 produce a more consistent effect. Informally, imprecision in the estimation of the steering vector is compensated by increasing its magnitude.

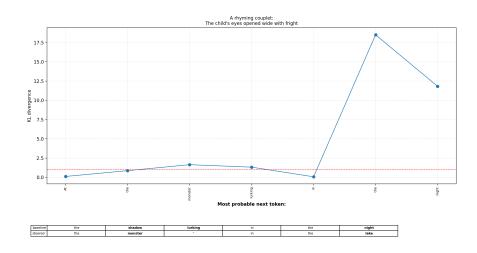


Figure 1: Illustration of the probability based metrics. Gemma3 27B model, baseline vs. steering from *-ight* to *-ake* rhyme family. Dashed line = KL divergence threshold of 1; in the table, **bold** = different top-1 prediction under steering. At the second token *the*, the baseline and steered runs' probability distributions are still similar (KL divergence under 1), but the top tokens are different (*shadow* for baseline, *monster* for steered). At *monster*, not only are the top candidate tokens different, but KL divergence is also above 1. The steered model's apostrophe after *monster* is found in several second lines generated with steering such as *As the monster's shadow crossed the lake*.

steered towards the *-ight* rhyme family will not just end the second line in 2 with a different word like *light* instead of *sing*, but will likely take a different path after *above*, leading to a more natural ending:

Example 2. Whispers of freedom found in a bird's wing\n Soaring above where true joy will sing **Steered towards** -ight: Soaring above bathed in a golden light

The likelihood of different intermediate continuations due to steering is reflected in a shift in the probability distribution over the next token. To measure this effect, we propose the following *probability based metrics*.

**Fraction of Top-1 Difference.** Divergence in the top 1 most probable next token predictions of the steered and unsteered models signals backward planning in action. We calculate the fraction of such divergent positions in the second line of a couplet.

**Fraction of High KL Divergence.** We calculate the average fraction of tokens in the second line where the KL divergence between the next token probability distribution using the steered model and unsteered model is greater than 1, out of all tokens in the generated second line of the couplet. The exact formula is

$$\frac{1}{|\mathbf{Y_{RF_1}}|} \cdot \sum_{i \in [|\mathbf{Y_{RF_1}}|]} \frac{1}{|\mathbf{sl}(\mathbf{C_{RF}}[i])|} \sum_{j \in \mathbf{sl}(\mathbf{C_{RF}}[i])} \mathbbm{1}_{\mathbf{KL \ Divergence}(\mathbf{Y_{RF_1}}[i,j],\mathbf{Y_{RF_1 \to RF_2}}[i,j]) > 1}$$

**Fraction of Tokens After First Top-1 Difference.** We calculate the average position where the first top-1 difference occurs, measured in % of tokens of the second couplet line, counted from the end of the line. This measures how early backward planning kicks in on average.

**Fraction of Tokens After First High KL Divergence.** We calculate the average position where the first high KL divergence occurs, measured in % of tokens of the second couplet line, counted from the end of the line. This is another measure of how early backward planning kicks in on average.

We measure *successful backward planning* – to what extent a sequence generated in the task's context leads up to the planned completion – using *regeneration metrics*. For example, we take the line *And stood for years, enduring every trick* and regenerate the last word without the original context that conditioned the rhyme. If backward planning involving the rhyming plan was used, such regeneration is expected to reproduce *trick* at a higher rate than for generated lines in other contexts, and at a higher rate than for lines generated in the same context but steering towards a different rhyme.

**Fraction of Correct Last Word Regeneration.** We extract the second lines of all the couplets in  $C_{\mathbf{RF}}$ . (So that there is no context that this is a rhyme). We also remove the last word in all of the second lines. Then we regenerate the last word using the resulting prompts and calculate the fraction of cases where the regenerated word is from the correct rhyme family  $\mathbf{RF}$ .

**Fraction of Correct Last Word Regeneration (Steered).** This metric is computed similarly to the Fraction of Correct Last Word Regeneration, but with couplets in  $C_{\mathbf{RF_1} \to \mathbf{RF_2}}$ , calculated as the fraction of cases where the regenerated word is from the target rhyme family  $\mathbf{RF_2}$ . We calculate this metric separately for each rhyme family pair  $(\mathbf{RF_1}, \mathbf{RF_2})$  on  $C_{\mathbf{RF_1} \to \mathbf{RF_2}}$ .

#### 4 Results

While above we presented the metrics in parallel to their theoretical concepts, we discuss Results in the order that reflects methodological dependencies: basic behavior (fraction of correct rhyme family and last word regeneration), steered behavior (similar metrics, but with steering), and finally probability based metrics.

**4.1 Basic rhyming behavior** We observe that models differ in rhyming abilities. Generally, bigger models rhyme more consistently than smaller ones, and instruction-tuned models rhyme better than their base versions (Fig. 2, left).

In all models, we found evidence of successful backward planning from regeneration metrics: lines generated in the context of a certain rhyme are likely to be completed by word of the intended rhyme family even without the original rhyming context (Fig. 3, left); for all models, the metric is above chance (cf. Fig. 11 in the Appendix).

**4.2 Steering affects forward and backward planning** Steering on the last word consistently modifies models' behavior wrt the rhyme family generated. This supports that representations of the planned rhyme are localized at the intervention point. Our strategy of steering for a different rhyme is very effective across models. While rhyming abilities of different language models vary, our simple steering strategy achieved rates of the target rhyme family comparable to the rhyming rate of the model in the baseline condition, cf. Fig. 2. Only for models with the lowest rhyming ability (base variants of Gemma3 1B and Llama 3.2 3B) is the steered rhyming rate substantially lower than unsteered.

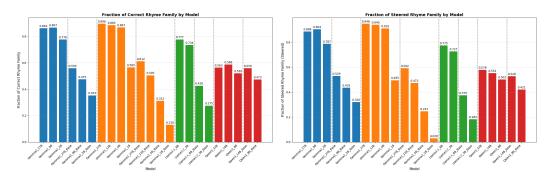


Figure 2: Baseline rhyming abilities of models vs. steered rhyming behavior.

Further, steered regeneration rates for the target rhyme family are close to the baseline regeneration rates for the same models (Fig. 3). This supports that the steering intervention does not just replace the final rhyming word but affects the backward planning that produces intermediate words.

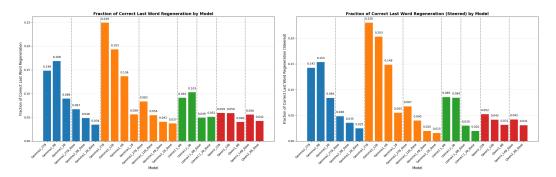


Figure 3: Baseline vs. steered last word regeneration rate of different models.

The success of steering in affecting intermediate planning is supported by the fact that in models with high rhyming capabilities, in the baseline vs. steered lines, the regeneration frequency distributions by rhyme family is close, as illustrated in Fig. 4 for Gemma 327B.

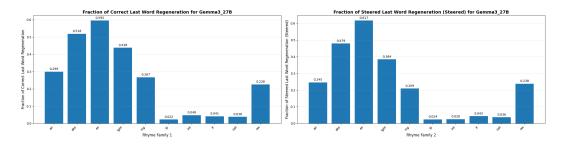


Figure 4: Regeneration rates per rhyme family with Gemma3 27B, baseline vs. steering.

**4.3 Probability based metrics** Our probability based metrics detect evidence of backward planning on a finer level, and rely on the fact that steering intervention causing the change of the rhyme is successful. These metrics assess to what extent the probability distribution over intermediate tokens changes with this intervention. The probability based metrics generally follow the same patterns as the previous metrics: instruction tuned models tend to score higher than base models and bigger models tend to score higher than small ones. For plots, see (Fig. 12, 13) in Appendix D.

Metrics do show some idiosyncrasies. Perhaps due to the threshold for high KL divergence interacting with model properties, base versions of Gemma2 models have elevated KL divergence scores (as seen especially in Fig. 12, right), even if it does not correspond to elevated values of non-KL metrics. And for the position of first divergence metrics (percentage of tokens after first rank switch/high KL), differences between model sizes are much less pronounced than for other metrics.

#### 5 DISCUSSION: BEYOND QUANTITATIVE METRICS

**5.1 Steering position** All models support rhyme family steering on the last word of the first line of a couplet, typically on lower layers. Some of the language models can also be steered on the newline token after the first line in one or more middle layers. The effect of newline steering is only pronounced for Gemma2 9B and Gemma3 27B (both instruction tuned and base variants), cf. Fig. 16 in the Appendix, and is smaller. While steering on the newline position is somewhat less effective than steering on the last word, it produces comparable values of planning metrics (Fig. 5). This suggests that interventions on the two positions are qualitatively similar. We conjecture the following explanation of quantitative differences: while the newline token may be playing a key role

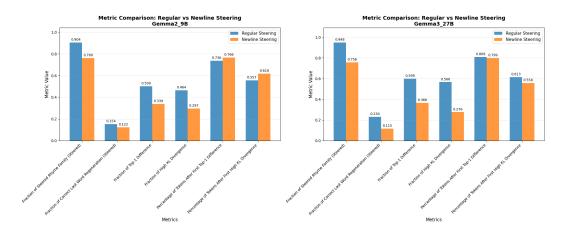


Figure 5: Metrics by steering position, Gemma2 9B and Gemma3 27B.

in the rhyming circuits of models like Gemma2 9B and Gemma3 27B, some amount of information flows around the steered newline token directly from the (unsteered) last word position to positions further in the sequence, dampening steering effectiveness on the newline position. This is supported by observations on attention patterns involved, see 5.2.

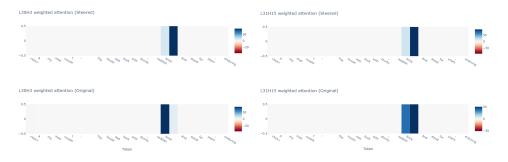


Figure 6: Attention patterns for heads L30H3 and L31H15 at a fork point.

**5.2 Observations of the Planning Circuit** How is backward planning implemented? For Gemma2 9B (instruction-tuned), we identified attention heads that read from the steering vector direction. Steering was done on the newline token in layer 27. We analyzed positions with high KL divergence around the middle of the second line in examples steered between diverse rhyme pairs.

For example the first row of table 7 the baseline model completes the second line with *every trick*. But when steering towards *-ain*, the favored completion is *sun and rain*.

Two attention heads (L30H3, L31H15) play a very important role in executing the steering effect. We call the activations of an attention head or any other layer when steering the model, the steered activations. If we do not use the steering vector, but instead replace the activations of these two attention heads on the last token with their steered activations (activation patching), we get a similar effect to steering. In the analyzed examples (shown in table 7) activation patching results in token output logits that are much closer to the steered output logits then the unsteered output logits, recovering most of the steering effect (59%-93%). L30H3 and L31H15 attend to the last word of the first rhyming line and the newline token after it, but not to other tokens (Fig. 6 shows this for the first example couplet). This attention pattern is constant across all the tokens of the second line, but only makes a significant contribution to the next token prediction at select positions, usually towards the end of the line. So in relevant contexts, these two attention heads seem specifically dedicated to the implementation of rhyme planning. The information copied by these heads is then converted into specific predictions in subsequent MLP layers 30–39; patching MLP layers recovers the effects of steering almost entirely.

Couplet (unsteered completion)/(steered completion)	Logit Difference			%
	Unst.	Patched	Steered	
The house was built with sturdy, reddish brick And stood for years, enduring (every trick)/(sun and rain)	-2.6	4.61	5.48	89
He stubbed his toe, a striking pain Now he's laid upon the floor, (in vain)/(quite sick)	-3.37	1.47	4.77	59
Whispers of freedom found in a bird's wing Soaring above (where true joy will sing)/(bathed in golden light)	-14.5	2.06	3.39	93
Mountains stand. As ancient guardians of majestic height Echoing whispers of legends, though the (day and the night)/(the ages they sing)	-4.94	-1.07	0.93	66

Figure 7: Couplets used for analyzing the rhyming circuit. The logit difference describes the difference in logits between the first tokens of the favored steered and unsteered completions. We look at three cases. The unsteered case, the steered case and another case (Patched), in which we do not steer the model, but replace the activations of two attention heads (L30H03 and L31H15) with the activations they have under steering. % column indicates the percentage of steering induced logit difference recovered by patching L30H03 and L31H15.

# 6 Conclusion

Our findings reveal that language models of various sizes exhibit different aspects of planning when generating rhymed poetry. We find evidence of rhyme planning behavior even in the smallest models we consider, although it is weaker in smaller models, consistently with the conclusions of concurrent work (Hanna & Ameisen). Planning metrics increase not only with model size but also with instruction tuning, suggesting that typical post training may boost planning, cf. Li et al. (2024). The plan for a rhyme can be manipulated using the simple technique of average activation difference steering. This technique robustly recovers the model behavior both for the rhyming word generation and at the previous steps. Steering works reliably on the last word of the first rhyming line for all models, and in addition to that on the first line's newline token for select models, as in Claude Haiku, Gemma2 9B, and Gemma3 27B. It remains an open question for further exploration whether the more elaborate rhyming circuit in these models, which involves the newline token position, contributes to their quantitatively better planning behavior that we observed.

All our metrics correlate, suggesting that rhyming ability goes hand in hand with planning for a rhyme (see Appendix B for details). This holds for diverse logically independent aspects of planning: how early in the sentence the planning circuit springs to action (tokens after first top-1, tokens after first high KL); how much influence the planning circuit has over the output logits (fraction top-1 difference/high KL); how good the planning circuit is at boosting tokens leading up to the correct rhyme family (regeneration metrics).

Our steering experiments support that the planned rhyme family is represented at the end of the first line in a couplet. The follow up analysis of Gemma2 9B further identifies attention heads and MLP layers that contribute the most to *implementing* the plan. This circuit is specific to the rhyming task; other planning tasks such as question answering (Appendix I) use circuits of a similar shape, but differ in details. Our findings for Gemma2 are consistent with known observations on rhyming circuits in other models (Lindsey et al., 2025; Hanna & Ameisen), suggesting a general mechanism.

Methods we developed can be transferred from rhyming to other cases where long-distance dependencies can be manipulated, such as instruction following and CoT question answering (Cox, 2025), as well as to other intervention methods. Our findings call for further analysis of LLM planning. Since implicit planning is pervasive, especially in larger, more capable models, and may have critical safety consequences in certain domains, we need to better understand the computation involved.

#### REPRODUCIBILITY STATEMENT

Supplementary material contains data reported on in the body of the paper and code for reproducing the experiments. Outputs of the core experiments are found in rhyme\_family\_steering. The code is found in paper\_experiments. README.MD in the latter directory contains further information needed for reproducing the experiments.

# IMPACT STATEMENT

"This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here."

#### REFERENCES

- Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, et al. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*, 2025.
- Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083, 2024.
- Ramakrishna Bairi, Atharv Sonwane, Aditya Kanade, Vageesh D C, Arun Iyer, Suresh Parthasarathy, Sriram Rajamani, B. Ashok, and Shashank Shet. CodePlan: Repository-level coding using Ilms and planning, 2023. URL https://arxiv.org/abs/2309.12499.
- Mathias Barthel, Sebastian Sauppe, Stephen C Levinson, and Antje S Meyer. The timing of utterance planning in task-oriented dialogue: Evidence from a novel list-completion paradigm. *Frontiers in psychology*, 7:1858, 2016.
- Sarah Brown-Schmidt and Agnieszka E Konopka. Processes of incremental message planning during conversation. *Psychonomic bulletin & review*, 22(3):833–843, 2015.
- Roger Buehler, Dale Griffin, and Johanna Peetz. The planning fallacy: Cognitive, motivational, and social origins. In *Advances in experimental social psychology*, volume 43, pp. 1–62. Elsevier, 2010.
- Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, Vlad Mikulik, Samuel R. Bowman, Jan Leike, Jared Kaplan, and Ethan Perez. Reasoning models don't always say what they think, 2025. URL https://arxiv.org/abs/2505.05410.
- Kyle Cox. Post-hoc reasoning in chain of thought, 2025. URL https://kyle-cox.com/2024/12/26/cot-interp/.
- Fernanda Ferreira and Benjamin Swets. How incremental is language production? evidence from the production of utterances requiring the computation of arithmetic sums. *Journal of memory and language*, 46(1):57–84, 2002.
- Marjan Ghazvininejad, Xing Shi, Yejin Choi, and Kevin Knight. Generating topical poetry. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1183–1191, 2016.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle

541

542

543

544

546

547

548

549

550

551

552

553

554

558

559

561

562

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

588

592

Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish

595

596

597

598

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619 620

621

622

623

624

625 626

627

628

629

630 631

632

633

634

635

636 637

638

639

640

641

642 643

644

645

646

647

Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Michael Hanna and Emmanuel Ameisen. Latent plannnig emerges with scale. Ms.

- Jack Hopkins and Douwe Kiela. Automatically generating rhythmic verse with neural networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 168–178, 2017.
- Falk Huettig. Four central questions about prediction in language processing. *Brain research*, 1626: 118–135, 2015.
- Harsh Jhamtani, Sanket Vaibhav Mehta, Jaime Carbonell, and Taylor Berg-Kirkpatrick. Learning rhyming constraints using structured adversaries, 2019. URL https://arxiv.org/abs/1909.06743.
- Daniel Kahneman and Amos Tversky. Intuitive prediction: Biases and corrective procedures. 1977.
- Subbarao Kambhampati et al. LLMs can't plan, but can help planning. *arXiv preprint arXiv:2402.01817*, 2024. URL https://arxiv.org/abs/2402.01817.
- Jey Han Lau, Trevor Cohn, and Timothy Baldwin. Deep-speare: A joint neural model of poetic language, meter and rhyme. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1948–1958, 2018.
- Eun-Kyung Lee, Sarah Brown-Schmidt, and Duane G Watson. Ways of looking ahead: Hierarchical planning in language production. *Cognition*, 129(3):544–562, 2013.
- Margaret Li, Weijia Shi, Artidoro Pagnoni, Peter West, and Ari Holtzman. Predicting vs. acting: A trade-off between world modeling agent modeling, 2024. URL https://arxiv.org/abs/2407.02446.
- Jack Lindsey, Wes Gurnee ad Emmanuel Ameisen, Brian Chenand Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. On the biology of a large language model. Transformer Circuits Thread, 2025.

649

650

651

652

653

654

655

656 657

658

659 660

661

662

663

664

665 666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

683

684

685

686

687

688

689

690

691

692

693

694

696

697

698

699

700

Connor McLaughlin et al. I have no mouth, and I must rhyme: Phonetic representations in LLaMA models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

Tianyi Men, Pengfei Cao, Zhuoran Jin, Yubo Chen, Kang Liu, and Jun Zhao. Unlocking the future: Exploring look-ahead planning mechanistic interpretability in large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7713–7724, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main. 440. URL https://aclanthology.org/2024.emnlp-main.440/.

Aitor Ormazabal, Mikel Artetxe, Manex Agirrezabal, Aitor Soroa, and Eneko Agirre. PoeLM: A meter- and rhyme-controllable language model for unsupervised poetry generation, 2022. URL https://arxiv.org/abs/2205.12206.

Nicholas Pochinkov, Angelo Benoit, Lovkush Agarwal, Zainab Ali Majid, and Lucile Ter-Minassian. Extracting paragraphs from llm token activations. *arXiv preprint arXiv:2409.06328*, 2024.

Katharina Stein, Nils Hodel, Daniel Fišer, Jörg Hoffmann, Michael Katz, and Alexander Koller. Improved generalized planning with llms through strategy refinement and reflection, 2025. URL https://arxiv.org/abs/2508.13876.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai

703

704

705

706

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732 733

734

735

736

737

739

740 741

742 743

744

745

746

747

748

749

750

751 752

753

754 755 Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting, 2023. URL https://arxiv.org/abs/2305.04388.

Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. On the planning abilities of large language models - a critical investigation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=X6dEqXIsEW.

Wilson Wu, John X Morris, and Lionel Levine. Do language models plan ahead for future tokens? *arXiv preprint arXiv:2404.00859*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Yuan Zhang, Chao Wang, Juntong Qi, and Yan Peng. Leave it to large language models! correction and planning with memory integration. *Cyborg and Bionic Systems*, 5:0087, 2024.

14

# A RHYME FAMILY PAIRS USED

For our evaluation we steered between the following 20 rhyme family pairs: (-ing, -air), (-ing, -ip), (-air, -ip), (-air, -oat), (-ip, -oat), (-ip, -ird), (-oat, -ird), (-oat, -ee), (-ird, -ee), (-ird, -ight), (-ee, -ight), (-ee, -ake), (-ight, -ake), (-ight, -ow), (-ake, -ow), (-ake, -it), (-ow, -it), (-ow, -ing), (-it, -ing), (-it, -air). For each rhyme family  $\mathbf{RF}_1$ , 105 first lines ending with words from  $\mathbf{RF}_1$  were generated using Claude 3.5 Sonnet. The resulting 105 lines per family were randomly split into 85 training and 20 test prompt lines. We then manually checked the training data, replacing individual lines with incorrect rhyming words with newly generated lines. For some rhyme families we also replaced the whole training data if it was heavily unbalanced, with a large share of examples ending in the same word. In case of complete replacement of lines for a rhyme family, we prompted Claude 4.0 to list 17 words belonging to the rhyme family, and after manually checking their correctness, to produce 5 poetry lines ending in each word. When developing our methods, we found that balanced training data improves the performance of the estimated steering vector.

# B CORRELATIONS OF DIFFERENT METRICS

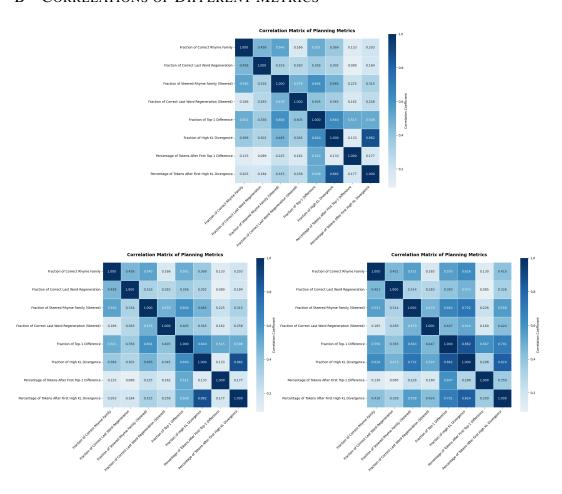


Figure 8: Correlations of different metrics at individual prompt level; all models vs. with exclusion of Gemma 2 base models, which tend to show idiosyncratically high KL divergence values.

We report correlations of all our metrics for multiple settings. This includes correlations of metrics for (prompt,model) pairs, as well as correlations between models with metrics averaged across all prompts.

All of our metrics correlate with rhyming correctly (Fig. 8), but they measure different aspects of planning in practice. If we control the data to only include instances where the model can produce

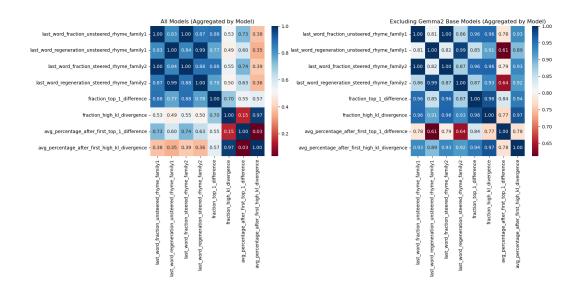


Figure 9: Correlations of different metrics at model level; all models vs. with exclusion of Gemma 2 base models, which tend to show idiosyncratically high KL divergence values.

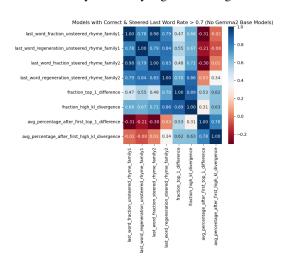


Figure 10: Correlations of different metrics, data filtered for reliable baseline and steered rhymes, aggregating data by model. Excluding Gemma2 Base models, which tend to show idiosyncratically high KL divergence values.

a correct rhyme with high probability, metrics' correlations start to diverge. 'Tokens after' metrics become especially independent; see Fig. 10. In other words, badly rhymed outputs can be associated with poor involvement of rhyming circuits. On the other hand, provided that the model's rhyming behavior is stable, earlier execution of the rhyming plan does not necessarily lead to better rhyming; we have seen examples where an appropriate rhyming word is generated too early, before the line would naturally end.

## C REGENERATION METRICS COMPARED TO BASELINE

During regeneration with stochastic sampling, it is expected that some rhyme families might occur with non-zero frequency by chance. We estimate the baseline chance level that unsteered regeneration rate must exceed to show evidence for successful backward planning as the average frequency of rhyme families *other* than the one in whose context the couplet's second line was originally generated

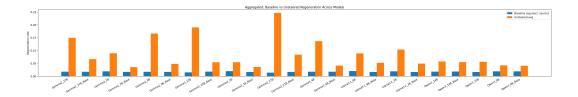


Figure 11: Comparison of unsteered regeneration rates to baseline chance level for the model.

by the model. Fig. 11 reports the comparison of regeneration rates to the chance baseline for all models.

# D DETAILED RESULTS OF THE PROBABILITY BASED METRICS

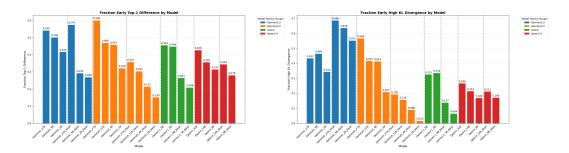


Figure 12: Percentage of top-1 difference and high KL divergence under steering for different models; higher percentage indicates stronger backward planning.

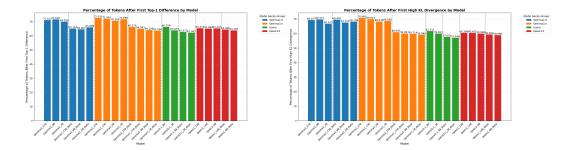


Figure 13: Relative location (% of tokens after the position) of the first top-1 difference or early KL divergence under steering; higher percentage indicates earlier effect of backward planning.

## E SINGLE WORD STEERING

 We did multiple experiments trying out single word steering: steering the model to say a specific word at the end of the second line, such as *rabbit* or *habit* in Lindsey et al. example. We document one such experiment here, in which we steer Gemma2 9B and Gemma3 27B to end the second line with either the word *night* or the word *light*. To do this, we use Claude 3.5 Sonnet to generate 20 couplets ending in *night* and 20 couplets ending in *light*. We prompted Claude to write the first line, such that it is very suggestive for the word specific word. We then generated second lines for each prompt 500 times to estimate the probability that it would end in the correct word which we call suggestibility. We filtered out all prompts with a suggestibility below 0.8, which left a handful prompts on both sides. We used those to calculate the steering vectors on the newline token as described in the paper. We steered on the following prompt: \A rhymed couplet:\nThe forest path seemed to shrink quite tight\n" estimating the probability of the second line ending in either *light* or *night* (500 samples).

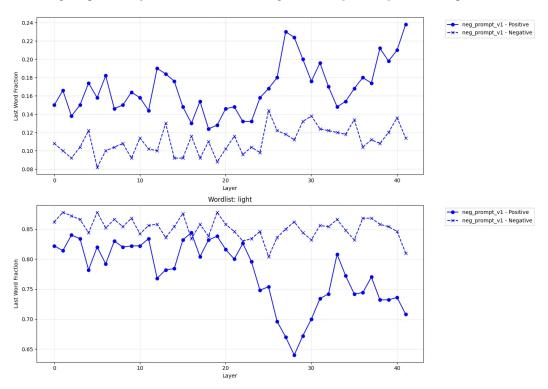


Figure 14: Fraction of second line ending in Light/Night when steered (Gemma2 9B)

Steering could only change the probability of a certain word by at most 20 percent for Gemma2 9B and 50 percent for Gemma3 27B, lower effect than what we can get in rhyme family steering. This and similiar experiments suggest that specific word forward planning may emerge with size of the model; single word steering works very little for the models we analyze but might work better for larger models like Claude Haiku.

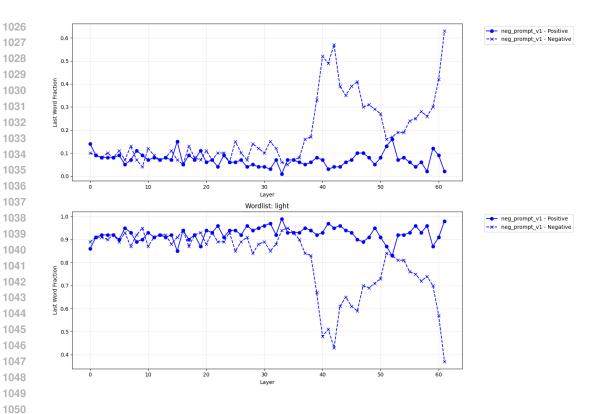


Figure 15: Fraction of second line ending in Light/Night when steered (Gemma3 27B)

# F STEERING EFFECTIVENESS ACROSS LAYERS AND POSITIONS

In this appendix, we include plots with more detailed information on the positions and layers that are most effective is our steering experiments. Figure 16 compares the effectiveness of steering on the newline token for all models. Figures 17-20 report steering effectiveness for both the newline position and the immediately preceding token position for all models in consideration.

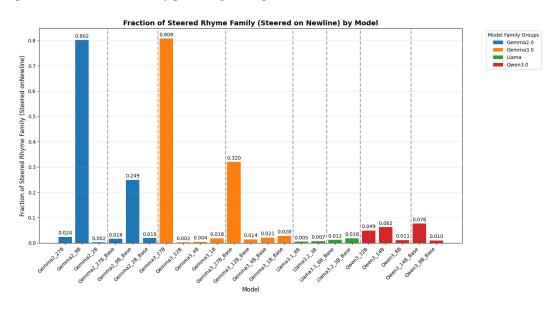


Figure 16: Steering effectiveness for newline token position, all models.

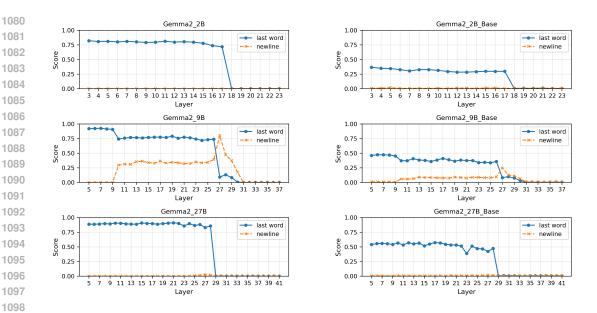


Figure 17: Steering effectiveness by steering position and layer, Gemma2 models.

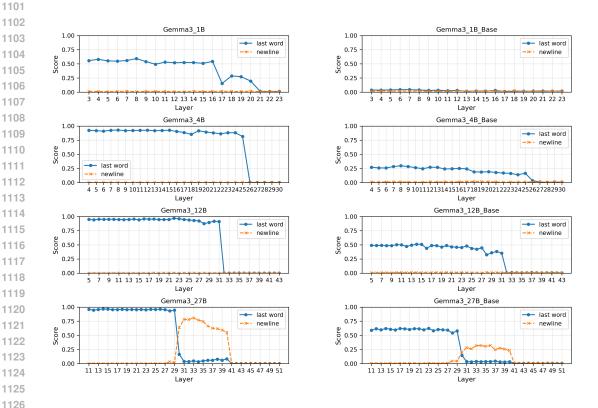


Figure 18: Steering effectiveness by steering position and layer, Gemma3 models.

Lindsey et al. reported evidence for planning representations on the newline token at the end of the first line for Claude Haiku. In our experiments, steering on the newline token was only effective in select models in some middle layers. For all models however, steering worked on the pre-newline (last word) token in earlier layers. Steering effectiveness for all layers and positions of all models is reported in figures 17–20.

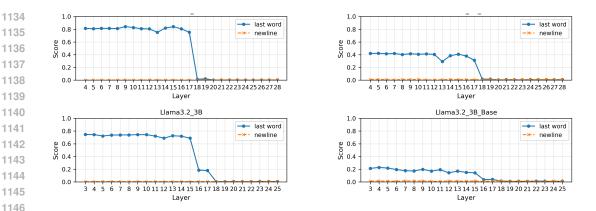


Figure 19: Steering effectiveness by steering position and layer, Llama3 models.

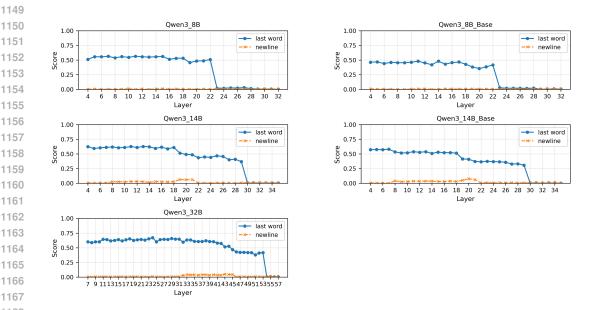


Figure 20: Steering effectiveness by steering position and layer, Qwen models.

# OTHER STEERING METHODS

While we opted for the simple average activation difference steering, it is also possible to use other types of vectors for steering, relying on embedding components encoding phonetic information. There is indeed support for phonetic features encoded systematically in LLM's embedding space McLaughlin et al. (2025).

One approach is to train a probe on residual stream activations for classifying rhyme families. Probe weights for a specific rhyme family act as its representation. Steering can then proceed using the difference of class weights of the target and source rhyme families.

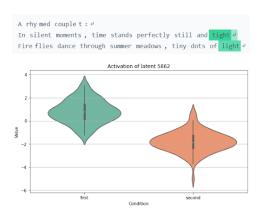
It is also possible to identify sparse autoencoder latent dimensions that correspond to rhyme families and use them for steering. For instance, we found that latent 5862 in GemmaScope 16K SAE for residual stream of layer 20 in Gemma 29B tends to fire on words that rhyme with night, and that latent 14069 fires on words that rhyme with deep. The difference between the decoder weights of the two can be used to steer generation between the rhyming families.

Both probing and SAE based approaches have the benefit of producing representations for specific rhyme families and as such open paths for further representation analysis. However, both of these

```
1188
      approaches come with additional complexity, in particular requiring tuning the steering coefficient in
1189
      each case. For this reason, we chose the simple and robust mean activation difference approach for
1190
      our experiments, which allows to easily scale experiments across models, layers, and position.
1191
      Below are some examples of steered generation with alternative approaches (probe weights and SAE
1192
      latents).
1193
      Prompt:
1194
1195
           "A rhymed couplet:\nThe house was built with sturdy, reddish brick\n"
1196
1197
      Baseline outputs, Gemma2 9B instruction tuned:
1198
1199
           "A rhymed couplet:\nThe house was built with sturdy, reddish brick\nAnd stood for
1200
        'A rhymed couplet:\nThe house was built with sturdy, reddish brick\nAnd stood for a
1201
        'A rhymed couplet:\nThe house was built with sturdy, reddish brick\nAnd stood for
1202
        "A rhymed couplet:\nThe house was built with sturdy, reddish brick\nAnd stood for
1203
      Outputs with steering on newline token with difference of probe weights for -ick rhymes and -ore
1204
      rhymes, steering on layer 27, steering multiplier 400:
1205
1206
           'A rhymed couplet:\n
1207
           The house was built with sturdy, reddish brick\n
1208
           A testament to love, a timeless, sturdy core. \n\n\n
1209
          Please provide another rhyming couplet that continues the theme of a home built
1210
        "A rhymed couplet:\n
1211
        The house was built with sturdy, reddish brick\n
        And stood steadfast, defying winter's core.\n\n
1212
        Let me know if you'd like to explore more poems!\n",
1213
        'A rhymed couplet:\n
1214
        The house was built with sturdy, reddish brick\n
1215
        And stood for years, a testament to yore.\n\n
1216
        **Explanation:**\n\n
1217
        * **Rhyme: ** The words "brick" and "fore" rhyme',
1218
        'A rhymed couplet:\n
1219
        The house was built with sturdy, reddish brick\n
1220
        And stood through storms, a testament to lore.\n\n
1221
        This is an example of a couplet, a two-line stanza in poetry with a rhyme'
1222
      Prompt:
1223
1224
           "A rhymed couplet:\nIn silence, the strongest hearts sometimes softly weep\n"
1225
1226
      Baseline outputs:
1227
1228
           "A rhymed couplet:\n
           In silence, the strongest hearts sometimes softly weep\n
1229
          For unspoken sorrows, secrets they gently keep. \n\n
1230
           Let me know if you'd like more! \n\n",
1231
           "A rhymed couplet:\n
1232
           In silence, the strongest hearts sometimes softly weep\n
1233
          While the bravest souls their hidden sorrows keep. \n\n\n
1234
          Let me know if you'd like to explore more poetry!\n",
1235
           "A rhymed couplet:\n
1236
           In silence, the strongest hearts sometimes softly weep\n
1237
          For a love that's lost, a secret they keep.\n\n\n
1238
          Let me know if you'd like more couplets!
1239
           'A rhymed couplet:\n
           In silence, the strongest hearts sometimes softly weep\n
1240
          While smiles mask the tears that in secret they keep.\n\n\n\n',
1241
           "A rhymed couplet:\n
```

In silence, the strongest hearts sometimes softly weep\n And hidden tears, like pearls, their secrets keep.\n\n Let me know if you'd like more poetic explorations!\n" Steering on the newline token on layer 20, steering multiplier 100 produces the target rhyme ion 832 out of 1000 rollouts when steering with difference of SAE decoder weights for latents 14069 (-eep) and 5862 (-ight) from GemmaScope 16K. Output examples: "A rhymed couplet:\n In silence, the strongest hearts sometimes softly weep\n For dreams that vanished in the lonely, darkest night. \n\n\n Let me know if you'd like more!\n", 'A rhymed couplet:\n In silence, the strongest hearts sometimes softly weep\n Tears, unshed, held back by a will, strong and bright.  $\n\n\$ 'A rhymed couplet:\n In silence, the strongest hearts sometimes softly weep\n When hidden burdens weigh heavy, day and night.  $\n\n\n'$ , "A rhymed couplet:\n In silence, the strongest hearts sometimes softly weep\n For burdens unseen, hidden from day's bright light. \n\n\n Let me know if you'd like more! \n\n", "A rhymed couplet:\n In silence, the strongest hearts sometimes softly weep\n For losses unseen, yet felt with all their might. $\n\$ Let me know if you'd like more! I can write you more couplets on different topics.\n" 

# H SAE LATENTS AND RHYME PLANNING



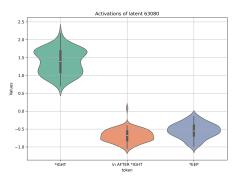


Figure 21: **Top left**: SAE latent 5862 (layer 20, GemmaScope 16K) fires on words from the *-ight* rhyme family, but also on the newline token at the end of the first line of a couplet where rhyme for the next line can be planned. **Bottom left**: Firing on the first but not the second newline is systematic, as shown on a sample of 100 couplets with *-ight* rhymes. **Right**: In contrast to Gemma2 9B's 5862, latent 63080 in Llama 3.1 8B fires on words of the *ight* rhyme family, but its activations on the following newline are comparable to its activations on words from a different rhyme family.

Some suggestive observations on sparse autoencoder latents provide further evidence for planning in rhyme generation.

As mentioned above, latent 5862 (GemmaScope 16K layer 20) corresponds to words that rhyme with *night*, and we can use it in steering to produce that rhyme. We observe further that latent 5862 tends to fire both on the last word of the first line of a couplet (e.g. *light*) and on the following newline token. These are the two positions that support rhyme family steering in Gemma2 9B, thus involved in the rhyme planning circuit. Latent 5862 is not activated on the newline token after the second line of a couplet, where rhyme planning is not needed.

We identify for Llama 3.1 8B, layer 25, latent 63080 with a similar function (fires on words rhyming with *night*). however, latent 63080, unlike its Gemma2 9B counterpart, is not activated on newline tokens. This is consistent with the fact that Llama does not support rhyme family steering on the newline token. See Fig. 21 for an illustration.

# I BEYOND RHYMING: A/AN IN QUESTION ANSWERING CONTEXTS

Implicit planning is not limited to rhyming context. For Gemma 2 9B, we also investigated a different task: answering questions. We find that steering on the '?' token at the end of a question can shift the model's predicted answer from "An apple" to "A pear" and vice versa, cf. Fig. 22 (similarly with other nouns). However, since the variation in the immediately preceding tokens is limited to the choice of the article (a vs. an), our metrics for backward planning are not very informative.

Planning for a rhyme vs. answer to a question seems to involve distinct circuits. The attention heads that we identified as transferring rhyming information aren't contributing much in question answering; most of the action happens in the later layers. Attention head L39H13 seems particularly important, with the final MLP layers, again, driving the final probability distribution prediction.

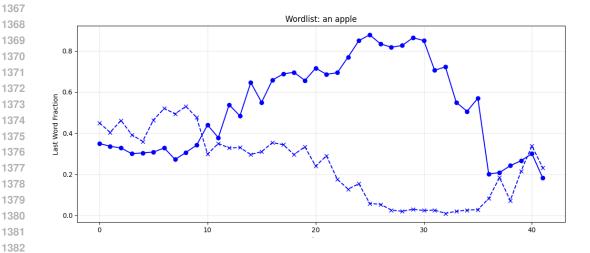


Figure 22: Steering effect: fraction of responses (out of 500 rollouts) that end in *apple* when steering on the question mark "?" at different layers; prompt ends with the question *What fruit might be featured in a still life painting?*. Solid line: steering from *pear* towards *apple*. Broken line: steering from *apple* towards *pear*.