ENHANCING LLM PLAN DIVERSITY BY TRAINING ON MULTIPLE PARTIAL-ORDER LINEARIZATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Plans comprise steps that may be executed in more than one order to reach a goal, since pairs of steps may not necessarily be dependent on each other. LLM planning has become increasingly important, along with models' ability to emit valid sequences of steps and remain robustly invariant to their valid reorderings. However, such invariance requires producing similar likelihoods under different valid orders of steps, which is in general not true of pre-trained LLMs. In this paper, we train LLMs to increase the reliability and diversity of the plans they generate. During training, we feed them fixed or multiple orders of the same plan, either using standard language modeling or a plan-unshuffling objective. Likelihood comparisons and sampled generation on plans in multiple domains show that training on a single linearization of a plan's partial order reduces diversity among generated valid orders. By contrast, using multiple valid orders can vastly increase diversity among generated valid orders, which in turn could improve the ability of LLMs to avoid infeasible execution paths and parallelize independent steps in workflows. Finally, we find that the activations and behavior of the model reflect the underlying structure of the plans, with limited evidence that the likelihoods of step orders can be steered with causal interventions on the activations. Code: `https://anonymous.4open.science/r/procedural_dags-1033`

1 INTRODUCTION

A plan specifies steps that lead from an initial to a goal state without necessarily having a unique valid order. In fact, the steps of a plan form a partial order and it may be possible for a subset of them to be permuted so that multiple step orders, i.e. linearizations, can lead to the same desired solution state while respecting the dependencies between steps. It is thus important for LLMs to behave invariantly in the way they model different alternative orders of a plan, since recognizing which steps must precede others and which can be reordered can support instruction-state tracking (Le et al., 2023), flexible plan execution (Muise et al., 2011), and parallel workflows (Kim et al., 2024).

However, training on unique step sequences rewards reproducing a specific order even if independent steps would allow for flexibility in the overall plan execution sequence. A model can therefore learn to prefer specific orders that impose unnecessarily strict dependencies between steps. Therefore, a model's capability of generating a valid order demonstrates that the constraints are satisfied, does not show whether it preserves the alternatives that the partial order permits. This raises the question as to how we can enhance the capability of LLMs to generate more diverse valid plans.

We do this by varying the number of valid linearizations shown to a model for each plan during training, from one fixed order to all distinct orders that respect the plan's partial-order constraints. We compare standard language modeling (LM) against a plan unshuffling objective, which involves reconstructing a valid order from a shuffled presentation of the same steps. The data we use includes the Blocks World domain (BW), whose explicit action rules support automatically verifiable plans (Valmeekam et al., 2023), and the human-written everyday procedures and annotated dependencies in proScript (Sakaguchi et al., 2021). We also use datasets comprising cooking recipes to examine whether the effects of our training scheme generalize to other domains.

We assess two factors. Assessing likelihoods lets us verify whether models prefer valid orders over alternatives that violate dependencies and how evenly they distribute the likelihoods across valid orders. Assessing sampled generations entails measuring how often their generated outputs

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

are valid and distinct. To understand how these behaviors relate to the underlying structure, we probe hidden activations for reachability between plan steps, which indicates whether one step must precede another. In addition, we test whether activation steering changes output ordering. These experiments examine whether greater output diversity is accompanied by more readily decodable plan structure, and whether the decoded features provide a means of influencing the model's choices.

Across all tested models on BW and proScript, training on multiple valid linearizations improves likelihood invariance relative to training on one fixed order, under both LM and unshuffling. On BW, unshuffling also yields more diverse generated valid orders, with most models retaining high validity. For GPT-2 and SmolLM-135M, much of the likelihood-diversity gain already appears with only a few valid training orders per plan. While generation on proScript remains less reliable and its diversity does not increase consistently with the number of training linearizations, the likelihood-diversity benefits extend from proScript unshuffling to recipe datasets.

On BW, unshuffling improves probe accuracy for incomparable step pairs, which can appear in either order. Increasing the number of training linearizations changes probe accuracy comparatively little despite its substantial effects on likelihood diversity. Probe-directed interventions on model activations affect likelihood preferences in some settings, but do not establish reliable control over generated step orders. Our findings show that exposing LLMs to alternative valid linearizations is an effective method of enhancing the diversity of plans they generate.

2 Related Work

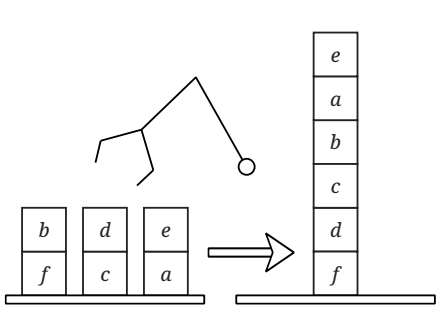
The planning literature encompasses a large number of domains, such as Blocks World (Slaney & Thiébaux, 2001; Valmeekam et al., 2023; Wang et al., 2026), cooking recipes (Yamakata et al., 2020; Bień et al., 2020; Lal et al., 2024; Diallo et al., 2024), everyday activities (Sakaguchi et al., 2021), and other procedural domains (Dalvi et al., 2018; Zhang et al., 2020; Jansen, 2020; Kim & Schuster, 2023; Zhang et al., 2024; Valmeekam et al., 2024; Lacombe et al., 2026).

Unshuffling objectives have already been used by Lee et al. (2020), who train BERT to unshuffle sentences for NLU, and by Nandy et al. (2024), who use BERT to classify recipe shufflings to improve entity tracking. However, while these approaches exploit ordering information, they do not consider invariance to valid reorderings, which is a fundamental aspect as regards the diversity of LLM plans, beyond just their validity. Plans in the Blocks World domain (Slaney & Thiébaux, 2001) provide explicit action rules; thus, their validity can be checked automatically, such as in PlanBench (Valmeekam et al., 2023). Despite this, valid outputs are not necessarily diverse: Wang et al. (2026) study a graph-based planning framework also applied to Blocks World and show that policy-gradient training can collapse solution diversity even after reaching perfect training accuracy. The problem of diversity is approached by Fredsgaard et al. (2026), who train autoregressive LLMs on alternative molecular-graph linearizations to make output likelihoods order-invariant. While our work is similar, we study DAGs underlying natural-language plans, comparing single fixed and multiple valid training orders under standard LM and plan unshuffling.

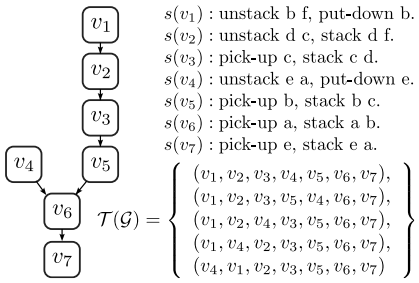
Beyond the effects of training on output behavior, probing (Belinkov, 2022) examines what structure models encode in their activations. While decodability does not establish whether a model uses the encoded information (Elazar et al., 2021), intervening on these representations can help assess their causal influence on model behavior. For example, Li et al. (2023) train a GPT model on Othello moves, decode game states from its activations, and steer those activations to change its next move. Extending this work, Vafa et al. (2024) develop metrics to evaluate the world model implicit in generated continuations. They find that accurate probing and next-token prediction can coexist with inconsistent continuations from prefixes reaching the same state. We similarly test whether procedural partial orders are decodable, whether activation interventions steer behavior, and how decodability varies with the number of training linearizations.

3 Background and notation

We represent a plan as a directed acyclic graph (DAG) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each node $v \in \mathcal{V}$ represents an action, described by a sequence of tokens $s(v)$. For distinct nodes $u, v \in \mathcal{V}$, we write $u \prec v$ if a directed path exists from u to v, meaning that u must precede v. These precedence constraints



(a) Illustration of the initial state and goal.



(b) Plan DAG $\mathcal{G}$, node descriptions, and the five valid linearizations $\mathcal{T}(\mathcal{G})$.

Figure 1: Blocks World example: six blocks in three towers must form one tower.

define a strict *partial order*. Two nodes are *comparable* if $u \prec v$ or $v \prec u$; otherwise, they are *incomparable*, written $u \parallel v$. We encode these relations in a *reachability matrix* R, with $R_{uv} = 1$ if $u \prec v$; 0 otherwise (i.e $u \parallel v$ implies $R_{uv} = R_{vu} = 0$).

A *linearization* π is a sequence of steps in which every node appears exactly once. It induces a *total order* through the positions of its nodes, including incomparable pairs. A linearization is valid if it respects all precedence constraints in $\mathcal{G}$. We denote the set of all valid linearizations by $\mathcal{T}(\mathcal{G})$. If every pair of distinct nodes is comparable, the partial order is a *chain*, meaning it admits exactly one valid linearization. If incomparable nodes exist, it admits multiple valid linearizations. We denote with $s(\pi) = \bigoplus_{v \in \pi} s(v)$ the token sequence of a linearization, where $\oplus$ denotes concatenation.

Figure 1 illustrates a plan in the BW domain, which comprises plans that instruct a robotic arm to organize blocks in a specific configuration. Node v_4 is incomparable with v_1, v_2, v_3, v_5, which form a chain, but must precede v_6 and v_7. It can therefore occupy any of the five positions before v_6, yielding five valid linearizations, one of which is $\pi = (v_1, v_2, v_3, v_4, v_5, v_6, v_7)$.

4 METHODS

Here, we describe our experimental design, driven by three research questions: **RQ1** (4.1.1): How does training on single vs multiple valid plan linearizations affect model likelihood invariance and generation diversity on plans? **RQ2** (4.1.2): Do LLMs encode plan structure in their activations through standard LM training, and can other objectives improve this encoding? **RQ3** (4.1.3): If present, does the encoded DAG information help the LLM generate more valid and diverse plans?

4.1 EXPERIMENTS

4.1.1 LANGUAGE MODELING AND UNSHUFFLING

To address **RQ1**, we vary training-order diversity under standard LM and plan unshuffling (Table 1). For LM, we train on valid linearizations with an LM loss on all tokens after BOS. For unshuffling, an invalid linearization is the *prefix*, followed by SEP and a valid *completion* of the same plan; the loss covers only tokens after SEP. This trains the model to unshuffle a plan, following Lee et al. (2020) and Nandy et al. (2024). An invalid linearization reverses at least one DAG precedence constraint: there exist steps $u \prec v$ such that v appears before u in π. Each objective has a control that uses invalid targets. In the unshuffling control, the invalid prefix and target are sampled independently.

For each plan $\mathcal{G}$, valid targets are sampled from a set of k valid serializations $\mathcal{T}_k(\mathcal{G}) \subseteq \mathcal{T}(\mathcal{G})$, where the latter is the set of all valid linearizations. We select this subset once per training run, with $k \in \{1, 2, 4, \text{all}\}$, where $k = \text{all}$ includes every valid linearization, capped at the number of available items. Each subset includes the valid order which is sampled when $k = 1$, (selection process described in Appendix A.1). On each visit to the plan, we sample a target linearization uniformly from this fixed subset. Thus, $k = 1$ always presents the same target. Larger values expose the model to k alternative valid orders of the same plan.

Table 1: Training settings. Highlighted tokens are covered by the loss. Valid orders are sampled from $\mathcal{T}_k(\mathcal{G})$. In the unshuffling control, invalid prefix π_{inv} and target π'_{inv} are sampled independently.

Setting	Single-sequence format	Setting	Prefix-completion format
LM	$[\text{BOS}, s(\pi_{\text{val}}), \text{EOS}]$	Unshuffling	$[\text{BOS}, s(\pi_{\text{inv}}), \text{SEP}, s(\pi_{\text{val}}), \text{EOS}]$
Control	$[\text{BOS}, s(\pi_{\text{inv}}), \text{EOS}]$	Control	$[\text{BOS}, s(\pi_{\text{inv}}), \text{SEP}, s(\pi'_{\text{inv}}), \text{EOS}]$

4.1.2 PROBING

To address **RQ2**, we probe a frozen LLM f for directed reachability (Elazar et al., 2021). To reduce probe positional shortcuts, we feed a randomly sampled invalid linearization π_{inv} to f and extract its token activations $H^{(L)} = f^{(L)}(s(\pi_{\text{inv}})) \in \mathbb{R}^{T\times d}$ at the last decoder block L, where T is the number of input tokens and d is the hidden dimension. We mean-pool each step's token activations into a representation $\mathbf{h}_u^{(L)}$. Following Conneau et al. (2018), given a pair (u, v) we use features $g(\mathbf{h}_u, \mathbf{h}_v) = \mathbf{h}_u \oplus \mathbf{h}_v \oplus (\mathbf{h}_u - \mathbf{h}_v) \oplus (\mathbf{h}_u \odot \mathbf{h}_v) \in \mathbb{R}^{4d}$ and fit a logistic regression probe q_ψ on them to predict R_{uv}. Feature equations and standardization are provided in Appendix A.3.

4.1.3 ACTIVATION STEERING

To address **RQ3**, we intervene on prefix-token activations for inputs in the unshuffling prefix-completion format and compare the resulting completion likelihoods and generations with those obtained without intervention. A reachability probe supplies logits z_{uv} and z_{vu} for the two directed relations between steps u and v. At decoder block l, we edit the token activations of one of these steps in the prefix to increase or decrease one logit while keeping the reverse logit fixed. We denote the intervention condition by $a \in \{uv^+, uv^-, vu^+, vu^-\}$, where uv and vu identify the target logit and the sign indicates an increase $(+)$ or decrease $(-)$ of the logit. We obtain the edit direction by projecting the target-logit gradient orthogonally to the gradient of the logit for the inverse direction, and then scale it by different doses. During the intervention, the LLM and probe parameters remain frozen. We compare against: 1) random-direction edits that measure responses to a general activation perturbation; 2) orthogonal edits that do not affect either of the probe logits; and 3) edits that use probes fitted on shuffled reachability labels. These comparisons assess whether the output response depends on the chosen probe direction. Appendix A.4 provides further details.

4.2 EVALUATION

4.2.1 ORDER LIKELIHOODS

Let $\ell_f(\pi)$ denote the summed token log probabilities of a linearization. We score tokens after BOS for LM settings and after SEP for unshuffling settings, using a fixed input prefix within each comparison for the latter (Table 1).

Validity. We denote with V_ℓ the fraction of pairs $(\pi_{\text{val}}, \pi_{\text{inv}})$ for which $\ell_f(\pi_{\text{val}}) > \ell_f(\pi_{\text{inv}})$. The invalid order is obtained by swapping nodes that are consecutive in the valid order and connected by a directed edge in the DAG, which breaks that edge's precedence constraint.

Diversity. For plans with more than one valid total order ($|\mathcal{T}(\mathcal{G})| > 1$), we form a probability distribution over all valid orders of each plan from their likelihoods and compute its normalized entropy. We denote the mean across these plans by D_ℓ; higher values indicate a more even distribution over valid linearizations. Appendix A.5.1 provides further details, along with supplementary bias and source-order preference metrics.

4.2.2 GENERATION

We sample two completions per test plan using the prefix-completion format, with a fixed shuffled linearization in the prefix to make the generation context identical. Details are in Appendix A.5.2.

Validity. Based on whether an output reproduces every input step exactly once and terminates after the final step, we classify it as *well-formed* or *malformed*. An output is classified as *valid* iff it is well-formed and respects all precedence constraints; otherwise, we classify it as an *invalid-order* output. We denote the fraction of valid outputs among all sampled outputs by V_{gen}.

Diversity. For each plan $\mathcal{G}$, let $n_{\mathcal{G}}$ be the number of valid sampled outputs and $d_{\mathcal{G}}$ the number of distinct valid linearizations among them. We measure diversity as the ratio $D_{\text{gen}} = \sum_{\mathcal{G}} d_{\mathcal{G}} / \sum_{\mathcal{G}} n_{\mathcal{G}}$ between distinct valid outputs and all valid outputs. Higher D_{gen} indicates less repetition.

4.2.3 PROBING

We evaluate the fitted probe on held-out plans. For each pair of distinct nodes u, v, the probe predicts reachability in both directions, R_{uv} and R_{vu}. We score these directed predictions separately across all evaluated plans, reporting accuracy[1] for comparable (A_{comp}) and incomparable (A_{inc}) node pairs. Accuracy on comparable pairs measures recognition of required constraints, while for incomparable pairs it measures recognition that the steps can appear in either order. Thus, we focus on A_{inc} in our discussion.

4.2.4 STEERING

On held-out plans, we select directly connected and incomparable node pairs (u, v) and construct candidate linearizations $[\pi_{\text{left}}, u, v, \pi_{\text{right}}]$ and $[\pi_{\text{left}}, v, u, \pi_{\text{right}}]$, with common prefix π_{left} and suffix π_{right}. The first linearization is valid, while the second breaks only the selected edge for directly connected nodes and remains valid for incomparable nodes. For both evaluations below the model receives a fixed context $c = [\text{BOS}, s(\pi_{\text{inv}}), \text{SEP}, s(\pi_{\text{left}})]$.

Likelihoods. The log-likelihood margin $M(a) = [\log p_f^a(s(u, v) \mid c) - \log p_f^a(s(v, u) \mid c)]/N$, where p_f^a is the distribution of model f under the intervention a and $N = |s(u)| + |s(v)|$, measures preference for uv over vu. Only u and v are scored; positive $M(a)$ favors uv regardless of the edited logit. We define the signed responses $\Delta M_{uv} = [M(uv^+) - M(uv^-)]/2$ and $\Delta M_{vu} = [M(vu^+) - M(vu^-)]/2$. Increasing z_{uv} should favor u before v ($\Delta M_{uv} > 0$), while increasing z_{vu} should favor v before u ($\Delta M_{vu} < 0$).

Generation. For each pair and intervention, we sample four completions from c. For a direct edge (u, v), let $n_e(a)$ count invalid-order outputs violating only that edge. Its target-edge violation rate is $e(a) = n_e(a)/4$ and $\Delta e = e(uv^-) - e(uv^+)$ is positive if decreasing z_{uv} increases these violations. Let $n_{uv}(a)$ and $n_{vu}(a)$ count valid outputs placing u before v and v before u, respectively. We define $P_{uv}(a) = n_{uv}(a)/4$, $P_{vu}(a) = n_{vu}(a)/4$, and $\Delta P(a) = P_{uv}(a) - P_{vu}(a)$. For incomparable nodes,[2] $\Delta P(a)$ measures the balance of valid relative orders, using *all* sampled outputs as the denominator: positive values favor u before v, while negative values favor the reverse. We compare edits to the two logits using $S^{\pm} = [\Delta P(uv^{\pm}) - \Delta P(vu^{\pm})]/2$. If generation follows the edited relation, we expect $S^+ > 0$: increasing z_{uv} favors u before v more than increasing z_{vu} does. Vice versa, with $S^- < 0$ decreasing z_{uv} favors v before u more than decreasing z_{vu} does. We average Δe and $S^{\pm}$ across plans, compare them with matched controls, and report validity shifts relative to the unedited baseline. Appendix A.5.3 provides further evaluation details.

4.3 DATA

We use DAG-annotated Blocks World plans (Slaney & Thiébaux, 2001) and everyday activities from proScript (Sakaguchi et al., 2021) for training with $k > 1$ valid linearizations per plan. RecipeNLG (Bień et al., 2020) provides training with a single observed order per recipe ($k = 1$). The English Recipe Flow Graph Corpus (ERFGC) (Yamakata et al., 2020) and PizzaCommonSense (PCS) (Diallo et al., 2024) are reserved for out-of-domain likelihood and reachability evaluation. Table 2 provides dataset sizes, splits, and graph properties.

BW and proScript. Our BW problems start with six blocks in three towers and have a single goal tower, as in Figure 1. Nodes represent block moves, with dependencies derived from the preconditions and effects of the actions. For BW we retain only plans with multiple valid orders. proScript has human-written steps and annotated dependencies, and retains both chains and plans with multiple valid orders.

RecipeNLG, ERFGC, and PCS. RecipeNLG lacks DAG annotations, so we impose its observed orders as chains and shuffle them for invalid-order controls. We obtain ERFGC DAGs by aggregat-

[1] Evaluating comparable pairs in both directions balances labels; incomparable-pair labels are all negative.

[2] For $u \prec v$, $P_{vu}(a) = 0$ and $\Delta P(a)$ equals the fraction of valid outputs.

Table 2: Dataset statistics. Steps are DAG nodes. Probe counts include both node-pair directions. RecipeNLG lacks dependency annotations, so we treat each recipe's observed step order as a chain.

	BW	proScript	ERFGC	RecipeNLG	PCS
Problems	30,000	4,337	271	2,030,282	538
Model-train split	24,001	3,252	–	2,028,282	–
Probe-train split	3,000	543	136	0	269
Probe-test split	2,999	542	135	0	269
Incomp. pairs #	14,576	792	518	0	8,704
Incomp. pairs %	10.2	3.1	21.7	–	39.0
Steps (mean; range)	7.4; 4–8	7.4; 7–9	4.3; 2–12	6.9; 2–71	9.1; 3–23
Chain share % ($\lvert\mathcal{T}(\mathcal{G})\rvert$=1)	0.0	67.4	46.1	100.0	8.4
Valid orders $\lvert\mathcal{T}(\mathcal{G})\rvert$ (median; max)	3; 18	1; 1,680	2; 336	1; 1	126; 9,900

Table 3: Experimental coverage. LM/U: language modeling/unshuffling; C: invalid-target controls. Only GPT-2 and SmolLM-135M cover $k = 2, 4$. *Five seeds for GPT-2 and SmolLM-135M; one otherwise. **GPT-2, SmolLM-135M, OPT-1.3B, and Qwen3-1.7B-Base. Pre-trained baselines are not evaluated on generation, since they cannot follow the unshuffling format without training.

Evaluation	Train → test	Models	Settings	k	Seeds
Probe / likelihood	BW → BW; proScript → proScript	9	LM/U+C	1,2,4,all	5
	proScript → ERFGC, PCS	9	LM/U+C	1,2,4,all	5*
	RecipeNLG → ERFGC, PCS, proScript	9	LM/U+C	1	1
Generation	BW → BW; proScript → proScript	9	U	1,2,4,all	5
Steering**	BW → BW	4	U+C	1,all	5

ing word-level dependencies at the step level, and PCS DAGs by heuristically matching annotated step inputs and outputs. PCS graphs can miss constraints, so their incomparability labels may include dependent pairs. We remove overlaps between both datasets and RecipeNLG before training.

Data splits. Blocks World and proScript have disjoint plans for LLM training (model-train split), probe fitting (probe-train split), and evaluation (probe-test split). For ERFGC and PCS, likelihood evaluation uses all recipes, while probe fitting and evaluation use disjoint halves (probe-train and probe-test, respectively). Appendix A.1 provides further information.

4.4 Models

We train the following pre-trained LLMs: three sizes of GPT-2 (124 M, 355 M, and 774 M parameters) (Radford et al., 2019), three sizes of SmolLM (135 M, 360 M, and 1.7 B parameters) (Allal et al., 2024), along with OPT-1.3B (Zhang et al., 2022), Pythia-1.4B (Biderman et al., 2023), and Qwen3-1.7B-Base (Yang et al., 2025). This lets us test a range of architectures at different sizes, using both absolute and relative positional embeddings (Su et al., 2024). We evaluate the final checkpoints. Appendix A.2 reports optimization settings and training budgets.

5 Results and Discussion

5.1 Likelihoods and generation

Order likelihoods. As Table 4a shows, training on alternative valid linearizations makes the likelihood distribution of the models less concentrated on specific orders. Across all nine models on both BW and proScript, $k = \text{all}$ yields higher D_ℓ than $k = 1$, for both LM and unshuffling. Relative to each objective's pre-trained baseline, $k = \text{all}$ raises D_ℓ for every model on BW. At $k = 1$, only SmolLM-135M and SmolLM-360M exceed the baseline. On proScript, LM generally remains below the baseline at both $k = 1$ and $k = \text{all}$. Using unshuffling, the two smaller SmolLMs represent an exception, surpassing the baseline already at $k = 1$ while gaining little at $k = \text{all}$. Still in the case of unshuffling with proScript, at $k = \text{all}$ only SmolLM-1.7B, Pythia, and Qwen remain below the baseline. Therefore, multiple training orders improve likelihood invariance relative to single-order training (**RQ1**), but do not always exceed the pre-trained baseline. The full results for BW and proScript are provided in Tables 7 and 10 in the appendix.

Table 4: Entropy D_ℓ over valid orders, averaged over five training seeds with ± 95% Student-t CI half-widths. Bold = highest per model and dataset within each panel.

(a) Entropy D_ℓ for all models at $k \in \{1, \text{all}\}$.

	Blocks World						proScript					
	LM			Unshuffling			LM			Unshuffling		
Model	Base	$k=1$	$k=\text{all}$	Base	$k=1$	$k=\text{all}$	Base	$k=1$	$k=\text{all}$	Base	$k=1$	$k=\text{all}$
GPT-2	.262	.178 ±.002	**.977** ±.001	.303	.111 ±.003	**.977** ±.003	**.656**	.242 ±.016	.299 ±.015	.343	.282 ±.010	.456 ±.014
GPT-2-medium	.268	.107 ±.004	**.980** ±.001	.294	.073 ±.005	.970 ±.003	**.641**	.150 ±.023	.237 ±.020	.390	.192 ±.021	.442 ±.019
GPT-2-large	.310	.059 ±.002	.976 ±.001	.313	.027 ±.003	**.984** ±.001	**.588**	.173 ±.016	.278 ±.020	.327	.206 ±.028	.464 ±.009
OPT-1.3B	.326	.060 ±.002	**.978** ±.001	.431	.086 ±.019	.957 ±.012	**.583**	.168 ±.035	.276 ±.030	.347	.180 ±.016	.476 ±.024
SmolLM-135M	.312	.517 ±.003	**.927** ±.001	.281	.642 ±.005	.861 ±.005	.574	.528 ±.010	.551 ±.001	.393	.730 ±.003	**.742** ±.004
SmolLM-360M	.353	.403 ±.002	**.934** ±.001	.389	.629 ±.003	.865 ±.004	.593	.546 ±.007	.558 ±.007	.460	.732 ±.003	**.753** ±.003
SmolLM-1.7B	.277	.039 ±.003	.952 ±.002	.405	.021 ±.002	**.986** ±.001	.549	.119 ±.017	.181 ±.010	**.566**	.195 ±.027	.495 ±.010
Pythia-1.4B	.270	.085 ±.009	**.971** ±.003	.416	.198 ±.070	.966 ±.003	**.571**	.160 ±.003	.264 ±.015	.466	.203 ±.018	.452 ±.035
Qwen3-1.7B-Base	.214	.040 ±.003	.980 ±.002	.541	.018 ±.002	**.989** ±.000	.578	.106 ±.011	.212 ±.031	**.722**	.173 ±.015	.556 ±.015

(b) Entropy D_ℓ for GPT-2 and SmolLM-135M at $k \in \{1, 2, 4, \text{all}\}$.

		Blocks World					proScript				
Model	Objective	Base	$k=1$	$k=2$	$k=4$	$k=\text{all}$	Base	$k=1$	$k=2$	$k=4$	$k=\text{all}$
GPT-2	LM	.262	.178 ±.002	.924 ±.003	.974 ±.002	**.977** ±.001	**.656**	.242 ±.016	.293 ±.013	.292 ±.013	.299 ±.015
	Unshuf.	.303	.111 ±.003	.937 ±.003	**.977** ±.001	**.977** ±.003	.343	.282 ±.010	.425 ±.025	.455 ±.010	.456 ±.014
SmolLM-135M	LM	.312	.517 ±.003	.895 ±.003	.925 ±.000	**.927** ±.001	.574	.528 ±.010	.546 ±.004	.548 ±.005	.551 ±.001
	Unshuf.	.281	.642 ±.005	.859 ±.003	.862 ±.002	.861 ±.005	.393	.730 ±.003	.737 ±.004	.739 ±.003	**.742** ±.004

Table 5: Generation validity V_ℓ and diversity D_ℓ at $k = 1$ and $k = \text{all}$ on Blocks World and proScript. Means ± 95% Student-t CI half-widths over five training seeds. Diversity is distinct-valid/valid, not coverage of all possible valid orders. Bold = highest per model, dataset, and metric. Invalid-output and malformed results in Tables 9 and 12.

	Blocks World				proScript			
	V_{gen}		D_{gen}		V_{gen}		D_{gen}	
Model	$k=1$	$k=\text{all}$	$k=1$	$k=\text{all}$	$k=1$	$k=\text{all}$	$k=1$	$k=\text{all}$
GPT-2	**.998** ±.002	.989 ±.009	.542 ±.013	**.828** ±.021	**.067** ±.008	**.067** ±.008	.692 ±.054	**.745** ±.032
GPT-2-medium	**.999** ±.003	.992 ±.004	.520 ±.008	**.837** ±.012	.097 ±.014	**.105** ±.017	.630 ±.027	**.656** ±.037
GPT-2-large	**1.000** ±.001	.990 ±.002	.512 ±.004	**.838** ±.008	**.098** ±.011	.094 ±.016	.620 ±.046	**.692** ±.038
OPT-1.3B	**.995** ±.004	.958 ±.047	.533 ±.011	**.829** ±.015	.118 ±.038	**.123** ±.029	.586 ±.041	**.668** ±.039
SmolLM-135M	.296 ±.013	**.368** ±.020	.875 ±.021	**.920** ±.019	**.009** ±.003	.005 ±.003	**1.000** ±.000	.933 ±.185
SmolLM-360M	.664 ±.038	**.783** ±.030	.794 ±.015	**.829** ±.026	**.015** ±.009	.014 ±.005	.913 ±.107	**1.000** ±.000
SmolLM-1.7B	**.998** ±.003	.995 ±.000	.510 ±.004	**.836** ±.008	.137 ±.025	**.167** ±.015	.588 ±.045	**.616** ±.024
Pythia-1.4B	**.996** ±.003	.995 ±.001	.573 ±.033	**.853** ±.007	.062 ±.009	**.066** ±.023	.577 ±.046	**.667** ±.069
Qwen3-1.7B-Base	**1.000** ±.000	.993 ±.003	.505 ±.003	**.817** ±.007	.337 ±.022	**.370** ±.010	.574 ±.025	**.634** ±.019

In Table 4b, at intermediate k values on BW, GPT-2 and SmolLM-135M show a large entropy increase at $k = 2$ under LM and unshuffling, approaching their $k = \text{all}$ values by $k = 4$. For example, GPT-2 unshuffling D_ℓ rises from .111 at $k = 1$ to .937 at $k = 2$ and .977 at $k = 4$. On proScript, GPT-2's entropy increases more under unshuffling than LM, while SmolLM-135M changes little under either objective. Thus, on BW, much of the observed entropy gain requires only a small subset of the available valid orders. Tables 8 and 11 report the full intermediate-k results.

With both LM and unshuffling, at $k = 1$ the validity V_ℓ is already almost perfect on BW (Table 7) and well above the pre-trained for proScript (Table 10). Increasing k mainly redistributes probability among valid alternatives. Expectedly, invalid-target controls have high entropy, but weak validity.

Generation. As shown in Table 5, sampling generations on BW confirms that training on multiple orders increases diversity. For most models, the fraction of valid outputs V_{gen} is high with both $k = 1$ and $k = \text{all}$. However, the diversity D_{gen} increses greatly when going from $k = 1$ to $k = \text{all}$. For example, SmolLM-1.7B's D_{gen} rises from .510 to .836 while validity stays near one. When going from $k = 1$ to $k = 2$ (Table 9), D_{gen} immediately jumps for GPT-2 ($0.542 \to 0.818$) and SmolLM-135M ($0.875 \to 0.900$). SmolLM-135M and SmolLM-360M are clear outliers, with much lower validity than the other models (in line with high D_ℓ at $k = 1$ as discussed for Table 4), but high diversity even when training with $k = 1$. Thus, validity and diversity must be considered jointly: while most of a model's valid outputs may be distinct, the overall fraction of valid plans may be low.

Table 6: Probe accuracy per incomparable-pair direction A_{inc} for $k \in \{1, \text{all}\}$ and invalid-target controls. averaged over five training seeds. Base = mean of five baseline-probe fits. Bold = highest per model/dataset. Confidence intervals are provided in the full results (Tables 7 and 10).

	Blocks World							**proScript**						
		LM			Unshuffling				LM			Unshuffling		
Model	Base	$k=1$	$k=$ all	Control	$k=1$	$k=$ all	Control	Base	$k=1$	$k=$ all	Control	$k=1$	$k=$ all	Control
GPT-2	.749	.748	.751	.790	.824	**.831**	.778	.569	.567	.574	.583	**.611**	.600	.598
GPT-2-medium	.750	.748	.755	.794	**.819**	.812	.779	.577	.570	.574	.568	**.593**	**.593**	.586
GPT-2-large	.755	.758	.765	.817	.837	**.843**	.794	.569	.554	.565	.570	**.580**	.579	.572
OPT-1.3B	.764	.750	.768	.828	.823	**.838**	.788	.552	.542	.552	.553	.554	**.557**	.545
SmolLM-135M	.748	.750	.750	.756	**.769**	.765	.760	.580	.583	.584	.579	.584	**.588**	.583
SmolLM-360M	.750	.750	.753	.766	**.781**	.777	.768	.588	.575	.569	.584	**.590**	.586	.575
SmolLM-1.7B	.748	.748	.764	.829	.826	**.852**	.783	.542	.539	.557	.546	.563	**.564**	.551
Pythia-1.4B	.750	.751	.764	.811	.811	**.831**	.795	.543	.541	.545	.546	.549	**.555**	.541
Qwen3-1.7B-Base	.752	.751	.768	.840	.838	**.860**	.786	.550	.534	.539	.540	.543	**.556**	.550

Generation results are weaker on proScript, with high shares of invalid and malformed outputs (full results in Table 12, Appendix B.3). For example, for GPT-2 with unshuffling $V_{\text{gen}} = .067$ at both k values. At $k = \text{all}$, Qwen has the highest validity, .370, with invalid orders still accounting for most outputs. Mean diversity increases for every model except SmolLM-135M, but the smallest SmolLMs have high D_{gen} alongside very low V_{gen}. On proScript, increasing k does not consistently increase mean generated diversity (GPT-2: $.692 \rightarrow .676 \rightarrow .701 \rightarrow .745$; SmolLM-135M: $1.000 \rightarrow .950 \rightarrow .900 \rightarrow .933$, for $k = 1, 2, 4, \text{all}$). As regards **RQ1**, training on multiple plan orders is therefore clearly beneficial for generation diversity on BW. On proScript, higher k also improves likelihood invariance and generation diversity, although not as consistently and with lower validity.

Out-of-domain evaluation. For every model trained with unshuffling on proScript, using $k = \text{all}$ valid orders per plan rather than $k = 1$ increases entropy D_ℓ on ERFGC (Table 13) and PCS (Table 15), consistent with the decrease in incomparable-swap likelihood bias. The intermediate-k sweeps show that, using unshuffling, GPT-2's D_ℓ increases when switching from $k = 1$ to $k = 2$ on both datasets, while SmolLM-135M changes little across k (Tables 14 and 16). Validity preference remains weak in many settings: at $k = \text{all}$, LM has higher V_ℓ than unshuffling for every model on PCS and for every model except OPT-1.3B and Qwen3-1.7B-Base on ERFGC. Relative to the baseline evaluated in the LM single-sequence format, proScript LM training usually lowers V_ℓ. Thus, the likelihood-diversity benefit transfers across domains even where validity preference remains weak.

When evaluating on ERFGC and PCS, training on RecipeNLG ($k = 1$) with LM increases V_ℓ for every model compared to the pre-trained baseline; by contrast, changes on proScript are mixed (Table 17). On the other hand, here LM also decreases entropy D_ℓ for every model on PCS, while changes on ERFGC and proScript are mixed, indicating substantial validity preference with concentrated likelihoods after single-order training. With unshuffling, training on RecipeNLG increases V_ℓ over the baseline for every model on ERFGC and all models except SmolLM-135M on PCS, while changes on proScript are mixed. Although LM generally has higher V_ℓ, unshuffling increases D_ℓ relative to its baseline for every model on proScript and for all models except Qwen3-1.7B-Base on ERFGC. On PCS, D_ℓ decreases except for SmolLM-135M and SmolLM-360M.

These results extend the answer to **RQ1** beyond the training domain: the likelihood-diversity benefit also transfers from proScript to ERFGC and PCS under unshuffling, even where validity preference remains weak. Furthermore, unshuffling can increase likelihood diversity even when training provides only one valid order per plan.

5.2 Structural decodability

As Table 6 shows, unshuffling at $k = \text{all}$ on BW yields higher incomparable-pair probe accuracy A_{inc} than valid-order LM for all nine models (A_{comp} results in Table 7). Its advantage over valid-order LM is smaller for SmolLM-135M and SmolLM-360M than for the remaining models. The control using invalid-order LM also improves A_{inc} over valid-order LM for every model, despite weak validity preference: Qwen3-1.7B-Base reaches $A_{\text{inc}} = .840$ but only has $V_\ell = .370$. These results address **RQ2**: reachability is decodable after LM training, and unshuffling improves incomparable-pair accuracy, but probe accuracy does not consistently track validity preference. Increasing k does not greatly change probe accuracy, including across the intermediate-k sweeps for

GPT-2 and SmolLM-135M (Tables 8 and 11). For GPT-2-medium and SmolLM-360M on BW, using unshuffling even decreases A_{inc} between $k = 1$ and $k = \text{all}$, while likelihood diversity increases.

On proScript at $k = \text{all}$, unshuffling's advantage over valid-order LM for A_{inc} is smaller than on BW for every model. Unshuffling and its invalid-target control yield similar mean A_{inc} values: for example, their paired difference for GPT-2 is $+.0015 \pm .0313$. In addition, using unshuffling boosts A_{comp} compared to valid-order LM for every model except SmolLM-360M (Table 10).

When evaluating on ERFGC, proScript training generally lowers probe accuracy relative to the pretrained baseline. For both LM and unshuffling, RecipeNLG training often improves comparable-pair accuracy, but these gains also occur under invalid-order LM training (Tables 13 and 17).

5.3 Activation steering

In some settings, logit edits can change the model's likelihood preference M between supplied step orders (Table 18). For GPT-2 trained with unshuffling at $k = \text{all}$, probe-directed and matched random edits do not differ significantly in how they change the likelihood preference for directly connected node pairs (the 95% CI for the difference in ΔM_{uv} includes zero). For SmolLM-135M trained with unshuffling at $k = 1$ or $k = \text{all}$, increasing the score for $u \to v$ (z_{uv}) on directly connected pairs favors u before v significantly more than decreasing it does ($\Delta M_{uv} > 0$). For SmolLM-135M in these two unshuffling settings, the probe-directed effect is significantly larger than the effects of random edits, orthogonal edits, or edits based on shuffled-label probes.

SmolLM-135M trained on invalid targets behaves in the opposite way on incomparable pairs: increasing z_{uv} favors v before u significantly more than decreasing it does ($\Delta M_{uv} < 0$). For OPT-1.3B and Qwen3-1.7B-Base, increasing rather than decreasing z_{uv} does not produce both a significant shift toward u before v ($\Delta M_{uv} > 0$) and a significantly larger shift than random edits in any of the tested training settings. For Qwen3-1.7B-Base under unshuffling at $k = 1$, increasing rather than decreasing the $v \to u$ score z_{vu} significantly favors v before u on directly connected pairs ($\Delta M_{vu} < 0$). This effect is significantly stronger than under random edits, but does not differ significantly from the effect of edits based on shuffled-label probes. These interventions offer only a partial answer to **RQ3**: probe-directed edits can affect output likelihoods in some settings, but do not establish that the model uses reachability information to choose step orders without intervention.

When generating continuations for incomparable pairs, editing z_{uv} rather than z_{vu} does not significantly shift the balance ΔP between valid outputs with u before v and those with v before u. This holds for both increasing (S^+) and decreasing (S^-) the scores in every trained model and setting, including comparisons with matched random edits. Decreasing rather than increasing z_{uv} does not significantly increase target-edge violations in any model or training setting. For SmolLM-135M under unshuffling at $k = 1$, however, Δe is significantly larger than under matched random edits. The generation tests therefore leave **RQ3** unresolved: they do not establish that the decoded features support more valid or diverse generated plans.

6 Conclusions

In this paper, we investigated how training on single versus multiple valid linearizations affects the diversity of generated plans. Under LM and unshuffling, training on multiple valid linearizations makes likelihoods more invariant across valid orders of held-out plans —the normalized entropy over valid orders is higher. On Blocks World, unshuffling also yields more diverse valid outputs, with most models retaining high validity; GPT-2 and SmolLM-135M achieve much of the likelihood-diversity gain with few training orders. On proScript, likelihood diversity improves and transfers to recipe datasets under unshuffling, but generation validity remains low and generated diversity does not increase consistently with k.

Reachability is decodable from activations, with unshuffling improving decodability most clearly on Blocks World. The gains in likelihood diversity from increasing training-order diversity are accompanied by comparatively small changes in probe accuracy. Steering affects likelihood preferences in some settings but does not establish control over generated orders. These findings support including alternative valid linearizations in training and evaluating validity and diversity together. Whether these alternatives improve adaptation to changing execution conditions remains to be tested.

AI USE STATEMENT

Initial drafting for this work is human-written, with subsequent iterative polishing with LLM assistance. All final paper content is human-edited. LLMs also assisted with coding and experiment orchestration, starting from initial human-only code and design. LLM-generated code was manually reviewed by the authors.

REPRODUCIBILITY STATEMENT

A repository with our code can be found at: `https://anonymous.4open.science/r/procedural_dags-1033`. Section 4 specifies the training conditions, model coverage, and metric definitions. Appendices A.3 to A.5 give the probe, intervention, and evaluation protocols.

REFERENCES

Loubna Ben Allal, Anton Lozhkov, and Elie Bakouch. SmolLM: Blazingly fast and remarkably powerful. Hugging Face technical blog, 2024. URL https://huggingface.co/blog/smollm.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022. doi: 10.1162/coli_a_00422. URL https://aclanthology.org/2022.cl-1.7/.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2397–2430. PMLR, 2023. URL https://proceedings.mlr.press/v202/biderman23a.html.

Michał Bień, Michał Gilski, Martyna Maciejewska, Wojciech Taisner, Dawid Wisniewski, and Agnieszka Lawrynowicz. RecipeNLG: A cooking recipes dataset for semi-structured text generation. In *Proceedings of the 13th International Conference on Natural Language Generation*, pp. 22–28. Association for Computational Linguistics, 2020. URL https://aclanthology.org/2020.inlg-1.4/.

Avrim L. Blum and Merrick L. Furst. Fast planning through planning graph analysis. *Artificial Intelligence*, 90(1–2):281–300, 1997.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data, July 2018. URL http://arxiv.org/abs/1705.02364. arXiv:1705.02364 [cs].

Bhavana Dalvi, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. Tracking State Changes in Procedural Text: a Challenge Dataset and Models for Process Paragraph Comprehension. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1595–1604, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1144. URL https://aclanthology.org/N18-1144/.

Aissatou Diallo, Antonis Bikakis, Luke Dickens, Anthony Hunter, and Rob Miller. PizzaCommonSense: A dataset for commonsense reasoning about intermediate steps in cooking recipes. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 12482–12496. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.findings-emnlp.728/.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021. doi: 10.1162/tacl_a_00359. URL https://aclanthology.org/2021.tacl-1.10/.

Laurits Fredsgaard, Aaron Thomas, Michael Riis Andersen, Mikkel N. Schmidt, and Mahito Sugiyama. Same graph, different likelihoods: Calibration of autoregressive graph generators via permutation-equivalent encodings, 2026. URL https://arxiv.org/abs/2604.05613.

Peter Jansen. Visually-Grounded Planning without Vision: Language Models Infer Detailed Plans from High-level Instructions. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4412–4417, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.395. URL https://aclanthology.org/2020.findings-emnlp.395/.

Najoung Kim and Sebastian Schuster. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3835–3855, 2023.

Sehoon Kim, Suhong Moon, Ryan Tabrizi, Nicholas Lee, Michael W. Mahoney, Kurt Keutzer, and Amir Gholami. An LLM compiler for parallel function calling. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 24370–24391. PMLR, 2024. URL https://proceedings.mlr.press/v235/kim24y.html.

Valentin Lacombe, Valentin Quesnel, and Damien Sileo. Reasoning Core: A Scalable Procedural Data Generation Suite for Symbolic Pre-training and Post-Training, March 2026. URL http://arxiv.org/abs/2603.02208. arXiv:2603.02208 [cs].

Yash Kumar Lal, Vanya Cohen, Nathanael Chambers, Niranjan Balasubramanian, and Ray Mooney. CaT-bench: Benchmarking language model understanding of causal and temporal dependencies in plans. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 19336–19354, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1077. URL https://aclanthology.org/2024.emnlp-main.1077/.

Duong Le, Ruohao Guo, Wei Xu, and Alan Ritter. Improved instruction ordering in recipe-grounded conversation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10086–10104. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.561. URL https://aclanthology.org/2023.acl-long.561/.

Haejun Lee, Drew A. Hudson, Kangwook Lee, and Christopher D. Manning. SLM: Learning a discourse language representation with sentence unshuffling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1551–1562. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.120. URL https://aclanthology.org/2020.emnlp-main.120/.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=DeG07_TcZvT.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization, January 2019. URL http://arxiv.org/abs/1711.05101. arXiv:1711.05101 [cs].

Christian Muise, Sheila A. McIlraith, and J. Christopher Beck. Monitoring the execution of partial-order plans via regression. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pp. 1975–1982, 2011. doi: 10.5591/978-1-57735-516-8/IJCAI11-330. URL https://www.ijcai.org/Abstract/11/330.

Abhilash Nandy, Yash Kulkarni, Pawan Goyal, and Niloy Ganguly. Order-based pre-training strategies for procedural text understanding. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 821–828, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-short.74. URL https://aclanthology.org/2024.naacl-short.74/.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL https://cdn.openai.com/better-language-models/language-models.pdf.

Keisuke Sakaguchi, Chandra Bhagavatula, Ronan Le Bras, Niket Tandon, Peter Clark, and Yejin Choi. proScript: Partially ordered scripts generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 2138–2149. Association for Computational Linguistics, 2021. URL https://aclanthology.org/2021.findings-emnlp.184/.

John Slaney and Sylvie Thiébaux. Blocks world revisited. *Artificial Intelligence*, 125(1-2):119–153, 2001. doi: 10.1016/S0004-3702(00)00079-5. URL https://linkinghub.elsevier.com/retrieve/pii/S0004370200000795.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing*, 568:127063, February 2024. ISSN 09252312. doi: 10.1016/j.neucom.2023.127063. URL https://linkinghub.elsevier.com/retrieve/pii/S0925231223011864.

Keyon Vafa, Justin Y. Chen, Ashesh Rambachan, Jon Kleinberg, and Sendhil Mullainathan. Evaluating the world model implicit in a generative model. In *Advances in Neural Information Processing Systems*, volume 37, pp. 26941–26975, 2024. doi: 10.52202/079017-0846. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/2f6a6317bada76b26a4f61bb70a7db59-Abstract-Conference.html.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change. *Advances in Neural Information Processing Systems*, 36:38975–38987, 2023.

Karthik Valmeekam, Kaya Stechly, and Subbarao Kambhampati. LLMs Still Can't Plan; Can LRMs? A Preliminary Evaluation of OpenAI's o1 on PlanBench, September 2024. URL http://arxiv.org/abs/2409.13373. arXiv:2409.13373 [cs].

Siwei Wang, Yifei Shen, Haoran Sun, Shi Feng, Shang-Hua Teng, Li Dong, Yaru Hao, and Wei Chen. BENEFITS AND PITFALLS OF REINFORCEMENT LEARNING FOR LANGUAGE MODEL PLANNING: A THEORETICAL PERSPECTIVE. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/01a8d63f9cb6dcbaa3092ccddd2075ac-Paper-Conference.pdf.

Yoko Yamakata, Shinsuke Mori, and John Carroll. English recipe flow graph corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 5187–5194. European Language Resources Association, 2020. URL https://aclanthology.org/2020.lrec-1.638/.

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Li Zhang, Qing Lyu, and Chris Callison-Burch. Reasoning about Goals, Steps, and Temporal Ordering with WikiHow. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4630–4639, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.374. URL https://aclanthology.org/2020.emnlp-main.374/.

Li Zhang, Hainiu Xu, Abhinav Kommula, Chris Callison-Burch, and Niket Tandon. OpenPI2.0: An Improved Dataset for Entity Tracking in Texts. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 166–178, St. Julian's, Malta, March 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.eacl-long.10. URL https://aclanthology.org/2024.eacl-long.10/.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. URL https://arxiv.org/abs/2205.01068.

A Additional methodological details

A.1 Data construction and splits

Blocks World. We construct problems with six blocks distributed among three initial towers and a single goal tower, and obtain one solution per problem with Graphplan (Blum & Furst, 1997). Each plan step groups the primitive actions from taking hold of a block on the table or another block to releasing it on the table or another block, so the robotic arm is empty at every step boundary. We derive dependencies from the steps' preconditions (what must hold before each step) and effects (how it changes the state), take the transitive reduction of the resulting DAG, and enumerate all its valid linearizations. We retain 30,000 plans with more than one valid linearization, so every retained plan contains incomparable steps. For $k = 1$, we use the solver's action order.

proScript. We use the human-written steps and annotated precedence graphs of proScript (Sakaguchi et al., 2021), with each step represented as one node. We take the transitive reduction of each graph and enumerate all valid linearizations. For $k = 1$, we select the valid linearization with minimum Kendall distance to the human draft order, breaking ties by the sequence of positions in that draft. This preserves the human order whenever it satisfies the annotated dependencies and repairs it otherwise. We retain both chains and plans with multiple valid linearizations.

RecipeNLG. RecipeNLG provides ordered cooking instructions (Bień et al., 2020). We represent each recipe's original step sequence as a chain for training with $k = 1$ and the corresponding invalid-order controls. Before constructing the training corpus, we remove detected overlaps with ERFGC and PCS using text and title matching. We exclude recipes with fewer than two distinct steps or whose paired input and completion exceed 1,024 tokens under the GPT-2 tokenizer, and hold out 2,000 recipes for validation.

Evaluation only. We use ERFGC and PCS for likelihood and reachability evaluation on plans in the cooking domain. For ERFGC (Yamakata et al., 2020), we project the annotated word-level flow relations onto the recipe steps to obtain a DAG. For PCS (Diallo et al., 2024), we derive dependencies by heuristically matching the annotated outputs of earlier steps to the inputs of later steps. These derived graphs can miss constraints, so their incomparability labels may include dependent step pairs. We retain the subset with fewer than 10,000 valid linearizations, allowing exhaustive enumeration. For both datasets, we pool the published splits, exclude recipes with fewer than two steps or paired representations exceeding the same GPT-2 context limit, and store transitively reduced DAGs and all their valid linearizations. Cyclic graphs are excluded from ERFGC.

Data splits. For Blocks World and proScript, we separate plans used for LLM training (model-train), probe fitting (probe-train), and evaluation (probe-test). Blocks World uses an approximately 80/10/10 split, stratified by the number of valid linearizations. For proScript, we preserve the published training split, divide the published development split into two approximately equal probe-fitting and evaluation subsets with the same stratification, and exclude the published test split. For ERFGC and PCS, likelihood evaluation uses all retained recipes, while reachability probes are fitted and evaluated on separate, approximately equal halves, again stratified by the number of valid linearizations.

A.2 Training hyperparameters

We use AdamW (Loshchilov & Hutter, 2019) as optimizer, with $\beta_1 = 0.9$, $\beta_2 = 0.95$, a peak learning rate of 10^{-4} and an effective batch size of 64. We set weight decay to 0.01 and gradient-norm clipping to 1.0. We linearly warm up the learning rate over the first 3% of training updates,

then apply cosine decay down to zero. The maximum input length into the models is 1,024 tokens. We train for 5k updates on BW, 2k on proScript, and 45k on RecipeNLG.

A.3 PROBING

Given a linearization π of a plan $\mathcal{G}$ and a frozen LLM f, we first extract contextual representations for tokens in π to obtain $H^{(l)} = f^{(l)}(s(\pi)) \in \mathbb{R}^{T\times d}$, where $l \in \{1, \dots, L\}$ indexes the LLM's decoder blocks, $H^{(L)}$ represents the last hidden states, d is the hidden dimension of the model, and T is the length of the sequence in tokens. Writing w_t for token t in $s(\pi)$, we compute step representations for node token sequences $s(u) = (w_{t_1}, \dots, w_{t_2})$ and $s(v) = (w_{t_3}, \dots, w_{t_4})$ of $\mathcal{G}$ by passing $H^{(l)}$ through a mean-pooling ϕ so that $\mathbf{h}_u^{(l)} = \phi(H^{(l)}_{t_1:t_2}) \in \mathbb{R}^d$ and $\mathbf{h}_v^{(l)} = \phi(H^{(l)}_{t_3:t_4}) \in \mathbb{R}^d$. Adapting the methodology of Conneau et al. (2018), we produce features $g(\mathbf{h}_u, \mathbf{h}_v)$ as:

$$g(\mathbf{h}_u, \mathbf{h}_v) = \mathbf{h}_u \oplus \mathbf{h}_v \oplus (\mathbf{h}_u - \mathbf{h}_v) \oplus (\mathbf{h}_u \odot \mathbf{h}_v) \in \mathbb{R}^{4d}\,,$$

where $\mathbf{h}_u \odot \mathbf{h}_v$ is the element-wise product (layer-index superscript omitted for readability). We fit a logistic regression probe q_ψ, with parameters $\psi = (\mathbf{w}, b)$, to estimate

$$q_\psi(R_{uv} = 1 \mid g(\mathbf{h}_u, \mathbf{h}_v)) = \sigma\big(\mathbf{w}^\top g(\mathbf{h}_u, \mathbf{h}_v) + b\big)\,,$$

where $\mathbf{w} \in \mathbb{R}^{4d}$, $b \in \mathbb{R}$, and σ is the sigmoid function.

Probe statistics. Prior to fitting, the features $g(\mathbf{h}_u, \mathbf{h}_v)$ fed into the probe are standardized using probe-training statistics, obtaining $\tilde{\mathbf{g}}_{uv}$. We fit the probe with ℓ_2 regularization (inverse regularization strength $C = 1$) and class weights inversely proportional to their frequencies in the probe-training data.

A.4 STEERING INTERVENTIONS

At decoder block l of the Transformer, we compute mean-pooled activations $\mathbf{h}_u^{(l)}$ and $\mathbf{h}_v^{(l)}$ as in Section 4.1.2 and then fit a probe q_ψ on $g(\mathbf{h}_u^{(l)}, \mathbf{h}_v^{(l)})$. The probe produces pre-sigmoid logits

$$z_{uv} = \mathbf{w}^\top g(\mathbf{h}_u, \mathbf{h}_v) + b\,, \quad z_{vu} = \mathbf{w}^\top g(\mathbf{h}_v, \mathbf{h}_u) + b\,,$$

which respectively represent $u \prec v$ and $v \prec u$. To isolate the effect of an intervention on the precedence $u \prec v$, we aim to modify z_{uv} while keeping z_{vu} fixed. To this end, let $\mathbf{a}_j = \nabla_{\mathbf{h}_j} z_{uv}$ and $\mathbf{b}_j = \nabla_{\mathbf{h}_j} z_{vu}$ be the gradients describing how the two scores respond to changes in the representation of step $j \in \{u, v\}$. In general, $\mathbf{a}_j^\top \mathbf{b}_j \neq 0$, so modifying z_{uv} could also increase z_{vu}. We therefore remove from $\mathbf{a}_j$ its component parallel to $\mathbf{b}_j$: $\mathbf{p}_j = \mathbf{a}_j - \frac{\mathbf{a}_j^\top \mathbf{b}_j}{\|\mathbf{b}_j\|_2^2}\mathbf{b}_j$, where if $\mathbf{b}_j = \mathbf{0}$ we simply use $\mathbf{p}_j = \mathbf{a}_j$. To modify z_{vu} while keeping z_{uv} fixed, we exchange the roles of $\mathbf{a}_j$ and $\mathbf{b}_j$ in the intervention construction. Finally, we scale this direction to obtain an activation intervention: $\boldsymbol{\delta}_j^\pm = \pm\tau \frac{\mathbf{p}_j}{\|\mathbf{p}_j\|_2^2}$, where $\tau > 0$ is the intervention dose. We add $\boldsymbol{\delta}_j^\pm$ to every input-token activation belonging to step j and we indicate the edited activations as $\widetilde{\mathbf{h}}_j^\pm = \mathbf{h}_j + \boldsymbol{\delta}_j^\pm$, where $\boldsymbol{\delta}_j^+$ increases z_{uv} by τ and the negative edit $\boldsymbol{\delta}_j^-$ lowers it by the same amount. At the selected dose τ, we choose $j \in \{u, v\}$ to minimize $\sqrt{|s(j)|}\,\|\boldsymbol{\delta}_j^\pm\|_2$, subject to a limit of 10% of that node's original activation norm.[3] If neither edit satisfies the bound, we skip the intervention. The LLM and fitted probe parameters remain frozen during intervention.

Intervention dose. We use three intervention doses $\tau \in \{0.5\sigma_z,\ \sigma_z,\ 2\sigma_z\}$, where σ_z is the standard deviation of the probe logits on the probe-fitting data. We use σ_z to make the intervention comparable across different probes.

Intervention layer choice. The layer l at which to apply the intervention for a specific LLM is chosen using the probe's reachability AUROC on development data. We then refit the probes on the full probe-training split before evaluating the interventions on testing data.

[3] This heuristic aims to limit broad disruption of the model's computations.

Controls. Random-direction edits measure how outputs respond to a general activation perturbation. Orthogonal edits use random directions projected to leave both real probe logits unchanged. Shuffled-label edits use directions derived from probes fitted after permuting node identities in the reachability labels. Each control edits the same step at the same block and matches the corresponding intervention's total activation norm. These comparisons assess whether the output response depends on the chosen probe direction beyond the magnitude of the perturbation.

A.5 Evaluation protocols

For results with five seeds, we report means and 95% Student-t confidence intervals with four degrees of freedom. Single-seed out-of-domain results are reported as point estimates. For steering, paired contrasts are computed within each seed before aggregation across seeds. Steering measurements are first averaged over plans within each seed; pre-trained steering baselines instead use problem-bootstrap confidence intervals.

A.5.1 Order likelihoods

For each plan $\mathcal{G}$, we normalize the likelihoods over valid linearizations $\pi \in \mathcal{T}(\mathcal{G})$ and compute their normalized entropy:

$$\tilde{p}_f(\pi \mid \mathcal{G}) = \frac{e^{\ell_f(\pi)}}{\sum_{\pi' \in \mathcal{T}(\mathcal{G})} e^{\ell_f(\pi')}}, \qquad D_\ell(f, \mathcal{G}) = -\frac{\sum_{\pi \in \mathcal{T}(\mathcal{G})} \tilde{p}_f(\pi \mid \mathcal{G}) \log \tilde{p}_f(\pi \mid \mathcal{G})}{\log |\mathcal{T}(\mathcal{G})|}.$$

Values near zero indicate concentration on one valid linearization, while one indicates uniformity. The reported likelihood diversity D_ℓ is the mean of $D_\ell(f, \mathcal{G})$ over plans with $|\mathcal{T}(\mathcal{G})| > 1$. The likelihood validity V_ℓ is the fraction of pairs $(\pi_{\text{val}}, \pi_{\text{inv}})$ for which $\ell_f(\pi_{\text{val}}) > \ell_f(\pi_{\text{inv}})$.

Additional likelihood metrics. For valid orders differing only by an adjacent incomparable-node swap, we define linearization bias as $\text{bias} = |\ell_f(\pi_{\text{val}}) - \ell_f(\pi'_{\text{val}})|/N$, where N is the common number of scored tokens. We average bias over all evaluated pairs in each dataset, with lower values indicating greater likelihood invariance. Source-order preference ("pref") is the fraction of pairs for which $\ell_f(\pi_{\text{val}}) > \ell_f(\pi'_{\text{val}})$, where the reference order π_{val} places the lower-indexed swapped node first.

For these metrics, we construct two valid linearizations from each selected pair of incomparable nodes, taking u to have the lower node index. We construct π_{left} as a valid linearization of all ancestors of either u or v, and π_{right} as a valid linearization of the remaining nodes, excluding u and v. Both orderings break ties using the smallest node index. The resulting linearizations are

$$\pi_{\text{val}} = [\pi_{\text{left}}, u, v, \pi_{\text{right}}], \qquad \pi'_{\text{val}} = [\pi_{\text{left}}, v, u, \pi_{\text{right}}].$$

A.5.2 Generation

For Blocks World and proScript, we sample two completions at temperature 1 for each of 200 held-out plans, using the unshuffling prefix-completion format with a fixed shuffled linearization per plan. We classify outputs and compute generation validity V_{gen} and diversity D_{gen} as defined in Section 4.2.

A.5.3 Steering

We evaluate the interventions described in Section 4.1.3 on held-out Blocks World plans.[4] For each selected plan, we choose a pair (u, v) linked by a direct edge or satisfying $u \parallel v$, and construct a valid linearization π' in which u immediately precedes v. Swapping these adjacent nodes produces π'', which remains valid for incomparable nodes and breaks only the edge (u, v) for directly connected nodes. We denote the common sequence preceding both nodes by π_{left} and the context for f as $c = [\text{BOS}, s(\pi_{\text{inv}}), \text{SEP}, s(\pi_{\text{left}})]$, where π_{inv} is a fixed invalid input linearization. We denote the intervention condition by $a \in \{uv^+, uv^-, vu^+, vu^-\}$, where uv^+ and uv^- respectively increase and decrease z_{uv}, and vu^+ and vu^- analogously affect z_{vu}.

[4] We evaluate on a subset to reduce compute time.

Likelihoods. We sample 256 plans for directly connected node pairs and 256 plans for incomparable pairs, and measure the preference of f between the two possible continuations of the selected steps:

$$M(a) = \frac{\log p_f^a(s(u) \oplus s(v) \mid c) - \log p_f^a(s(v) \oplus s(u) \mid c)}{|s(u)| + |s(v)|},$$

where p_f^a is the model's output distribution under that intervention. A positive $M(a)$ indicates a preference for placing u before v. We define the signed likelihood responses as

$$\Delta M_{uv} = \frac{M(uv^+) - M(uv^-)}{2}, \qquad \Delta M_{vu} = \frac{M(vu^+) - M(vu^-)}{2}.$$

Since $M(a)$ always measures log-likelihood of uv minus log-likelihood of vu, regardless of which logit we edit, positive ΔM_{uv} and negative ΔM_{vu} indicate that the model favors each continuation more when its corresponding probe logit is increased than when it is decreased. We report mean ΔM across evaluated plans.

Generation. We sample 128 plans for directly connected node pairs and 128 for incomparable node pairs. For each plan, we sample four continuations per intervention type a at temperature 1. We prepend π_{left} before evaluating each complete output.

For directly connected nodes, let $n_e(a)$ count outputs that are invalid only because they break the edge $(u, v) \in \mathcal{E}$, while satisfying the other validity requirements. We define $e(a) = n_e(a)/4$ and measure the effect of weakening versus strengthening the corresponding probe logit as $\Delta e = e(uv^-) - e(uv^+)$.

For incomparable nodes, let $n_{uv}(a)$ and $n_{vu}(a)$ count valid outputs placing u before v or v before u, respectively. We define $P_{uv}(a) = n_{uv}(a)/4$, $P_{vu}(a) = n_{vu}(a)/4$, and $\Delta P(a) = P_{uv}(a) - P_{vu}(a)$. We compare interventions on the two directions using $S^{\pm} = [\Delta P(uv^{\pm}) - \Delta P(vu^{\pm})]/2$. Positive $\Delta P(a)$ indicates more valid outputs with u before v, while negative values indicate the reverse. The contrasts compare increasing z_{uv} with increasing z_{vu} (S^+), and decreasing z_{uv} with decreasing z_{vu} (S^-). If increasing a probe logit shifts the balance of valid outputs toward the corresponding step order, and decreasing it shifts the balance toward the reverse order, we expect $S^+ > 0$ and $S^- < 0$.

We average these contrasts across evaluated plans and compare the effects of these interventions with those of the matched controls described in Appendix A.4. We also report changes in validity relative to the unedited baseline.

B Supplementary results

B.1 Reading the full result tables

The full likelihood and probing tables group settings under LM and Unshuffling. Within each group, $k = 1$ uses one fixed valid target and $k = \text{all}$ samples from all valid targets. The Control columns use invalid targets. The pretrained baseline is shown separately on each side: its probe results are repeated, while its likelihood results use the corresponding LM or unshuffling input format. Intermediate $k = 2, 4$ conditions cover GPT-2 and SmolLM-135M. Bold marks maxima across the displayed settings for each dataset, model, and metric, including bases, controls, and ties at the printed precision. For bias, invalid-order rates, and malformed rates, minima are marked; for pref., values closest to .5 are marked. Steering follows the predicted directions specified in its caption.

In these tables, A_{inc} denotes no-path recall over directional examples from incomparable node pairs; A_{comp} is accuracy over directional examples from comparable pairs; V_ℓ denotes likelihood validity, the fraction of edge-breaking serialization pairs for which the valid order has higher likelihood; D_ℓ denotes likelihood diversity, the mean normalized entropy over valid orders; bias is incomparable-swap likelihood bias; and pref. is incomparable-swap source preference, with .5 denoting equal aggregate preference. High entropy and low bias describe likelihoods among valid alternatives and must be read alongside likelihood validity V_ℓ. In the generation tables, V_{gen} is the valid-output fraction and D_{gen} is the distinct-valid/valid ratio. Where intervals are available, entries are means $\pm$ 95% Student-t confidence-interval half-widths across five training seeds. The point estimates for single-seed settings do not quantify training-seed uncertainty.

B.2 BLOCKS WORLD

Table 7: Blocks World results at update 5,000 for all nine models and $k \in \{1, \text{all}\}$. All trained settings use five seeds. Column and metric definitions are given at the start of Appendix B.

		LM				Unshuffling			
	model	base	$k=1$	$k=\text{all}$	Control	base	$k=1$	$k=\text{all}$	Control
$A_{\text{inc}} \rightarrow$	gpt2	.749±.002	.748±.001	.751±.006	.790±.008	.749±.002	.824±.002	**.831±.005**	.778±.004
	gpt2-medium	.750±.003	.748±.002	.755±.005	.794±.007	.750±.003	**.819±.003**	.812±.011	.779±.005
	gpt2-large	.755±.002	.758±.005	.765±.003	.817±.007	.755±.002	.837±.005	**.843±.002**	.794±.007
	opt-1.3b	.764±.002	.750±.004	.768±.003	.828±.005	.764±.002	.823±.010	**.838±.006**	.788±.005
	SmolLM-135M	.748±.002	.750±.002	.750±.001	.756±.001	.748±.002	**.769±.001**	.765±.001	.760±.002
	SmolLM-360M	.750±.003	.750±.002	.753±.002	.766±.003	.750±.003	**.781±.004**	.777±.005	.768±.002
	SmolLM-1.7B	.748±.002	.748±.004	.764±.003	.829±.009	.748±.002	.826±.007	**.852±.005**	.783±.008
	pythia-1.4b	.750±.002	.751±.005	.764±.004	.811±.011	.750±.002	.811±.011	**.831±.011**	.795±.003
	Qwen3-1.7B-Base	.752±.002	.751±.003	.768±.007	.840±.005	.752±.002	.838±.006	**.860±.006**	.786±.007
$A_{\text{comp}} \rightarrow$	gpt2	.835±.002	.833±.004	.843±.003	**.890±.002**	.835±.002	.887±.002	.889±.003	.840±.003
	gpt2-medium	.834±.001	.831±.001	.845±.002	**.895±.002**	.834±.001	.873±.002	.879±.001	.850±.004
	gpt2-large	.836±.002	.840±.002	.847±.002	**.900±.002**	.836±.002	.882±.002	.884±.001	.847±.004
	opt-1.3b	.840±.002	.833±.002	.849±.002	**.897±.001**	.840±.002	.860±.007	.878±.006	.843±.004
	SmolLM-135M	.830±.002	.835±.001	.837±.002	.851±.001	.830±.002	**.865±.000**	**.865±.002**	.846±.002
	SmolLM-360M	.836±.002	.835±.001	.843±.001	.861±.001	.836±.002	**.879±.002**	.878±.001	.850±.002
	SmolLM-1.7B	.833±.001	.828±.001	.841±.002	**.900±.002**	.833±.001	.883±.003	.892±.003	.849±.007
	pythia-1.4b	.834±.002	.827±.003	.840±.002	**.894±.002**	.834±.002	.855±.011	.873±.004	.847±.002
	Qwen3-1.7B-Base	.835±.002	.834±.002	.848±.003	**.903±.001**	.835±.002	.885±.004	.897±.003	.850±.003
$V_\ell \rightarrow$	gpt2	.742	.966±.003	**1.000±.000**	.533±.025	.523	.997±.001	**1.000±.000**	.536±.015
	gpt2-medium	.714	.970±.003	**1.000±.000**	.478±.047	.513	.994±.002	**1.000±.000**	.473±.053
	gpt2-large	.723	.988±.003	**1.000±.000**	.420±.021	.546	.994±.001	**1.000±.000**	.490±.025
	opt-1.3b	.664	.980±.005	**1.000±.000**	.437±.049	.560	.990±.002	**1.000±.000**	.497±.011
	SmolLM-135M	.814	**1.000±.000**	**1.000±.000**	.587±.019	.651	.982±.001	.996±.000	.494±.006
	SmolLM-360M	.750	.998±.000	**1.000±.000**	.555±.008	.578	.979±.002	.995±.001	.497±.005
	SmolLM-1.7B	.726	.982±.001	**1.000±.000**	.404±.025	.520	.994±.001	**1.000±.000**	.477±.026
	pythia-1.4b	.736	.974±.008	**1.000±.000**	.418±.049	.595	.993±.002	**1.000±.000**	.463±.015
	Qwen3-1.7B-Base	.825	.988±.004	**1.000±.000**	.370±.027	.521	.994±.001	**1.000±.000**	.454±.012
$D_\ell \rightarrow$	gpt2	.262	.178±.002	.977±.001	.990±.001	.303	.111±.003	.977±.003	**.999±.000**
	gpt2-medium	.268	.107±.004	.980±.001	.988±.002	.294	.073±.005	.970±.003	**.999±.000**
	gpt2-large	.310	.059±.002	.976±.001	.986±.002	.313	.027±.003	.984±.001	**.999±.000**
	opt-1.3b	.326	.060±.002	.978±.001	.983±.002	.431	.086±.019	.957±.012	**.998±.000**
	SmolLM-135M	.312	.517±.003	.927±.001	.960±.002	.281	.642±.005	.861±.005	**.988±.001**
	SmolLM-360M	.353	.403±.002	.934±.001	.943±.001	.389	.629±.003	.865±.004	**.986±.000**
	SmolLM-1.7B	.277	.039±.003	.952±.002	.975±.004	.405	.021±.002	.986±.001	**1.000±.000**
	pythia-1.4b	.270	.085±.009	.971±.003	.982±.002	.416	.198±.070	.966±.003	**.998±.000**
	Qwen3-1.7B-Base	.214	.040±.003	.980±.002	.977±.001	.541	.018±.002	.989±.000	**.998±.000**
bias $\leftarrow$	gpt2	.0711	.1767±.0050	.0040±.0001	.0024±.0002	.0599	.0971±.0021	.0041±.0004	**.0008±.0001**
	gpt2-medium	.0670	.2551±.0217	.0038±.0001	.0025±.0003	.0618	.1470±.0060	.0045±.0003	**.0006±.0001**
	gpt2-large	.0618	.2247±.0122	.0040±.0001	.0026±.0003	.0595	.1853±.0081	.0035±.0001	**.0005±.0000**
	opt-1.3b	.0563	.2610±.0201	.0040±.0001	.0031±.0002	.0397	.1868±.0177	.0055±.0012	**.0011±.0000**
	SmolLM-135M	.0564	.0307±.0001	.0080±.0001	.0054±.0001	.0583	.0242±.0002	.0117±.0004	**.0030±.0001**
	SmolLM-360M	.0473	.0574±.0018	.0079±.0001	.0067±.0001	.0421	.0254±.0002	.0112±.0004	**.0032±.0000**
	SmolLM-1.7B	.0564	.3912±.0159	.0058±.0001	.0035±.0002	.0414	.2754±.0193	.0032±.0001	**.0004±.0000**
	pythia-1.4b	.0621	.3342±.0305	.0048±.0002	.0032±.0002	.0422	.1307±.0133	.0048±.0002	**.0012±.0001**
	Qwen3-1.7B-Base	.0862	.4311±.0357	.0042±.0001	.0042±.0001	.0320	.2825±.0196	.0032±.0001	**.0014±.0001**
pref. $\rightarrow .5$	gpt2	.506	.729±.003	**.499±.011**	.512±.013	.509	.716±.005	.441±.008	.483±.048
	gpt2-medium	**.500**	.736±.008	.501±.010	.528±.021	.512	.690±.012	.428±.011	.519±.032
	gpt2-large	.471	.761±.004	**.497±.010**	.548±.018	.505	.730±.006	.448±.010	.510±.037
	opt-1.3b	.519	.756±.006	**.504±.008**	.523±.012	.492	.671±.016	.447±.015	.505±.007
	SmolLM-135M	.532	.666±.005	.519±.004	.489±.014	**.490**	.611±.002	.473±.005	.511±.003
	SmolLM-360M	.493	.658±.001	.514±.007	**.497±.010**	.476	.603±.007	.451±.005	.506±.004
	SmolLM-1.7B	.514	.743±.005	**.495±.009**	.517±.022	.486	.711±.004	.453±.009	.523±.032
	pythia-1.4b	.521	.726±.009	**.499±.011**	.513±.004	.489	.644±.025	.430±.009	.504±.021
	Qwen3-1.7B-Base	.519	.756±.005	.495±.012	.508±.009	**.501**	.703±.011	.457±.013	**.499±.011**

Table 8: Blocks World results at update 5,000 for GPT-2 and SmolLM-135M with $k \in \{1, 2, 4, \text{all}\}$. All trained settings use five seeds; columns and metrics follow Appendix B.

		LM						Unshuffling					
	model	base	$k=1$	$k=2$	$k=4$	$k=\text{all}$	Control	base	$k=1$	$k=2$	$k=4$	$k=\text{all}$	Control
$A_{\text{inc}} \uparrow$	gpt2	.749$_{\pm.002}$	.748$_{\pm.001}$	.751$_{\pm.002}$	.751$_{\pm.004}$	.751$_{\pm.006}$	.790$_{\pm.008}$	.749$_{\pm.002}$	.824$_{\pm.002}$	.830$_{\pm.005}$	**.832$_{\pm.005}$**	.831$_{\pm.005}$	.778$_{\pm.004}$
	SmolLM-135M	.748$_{\pm.002}$	.750$_{\pm.002}$	.750$_{\pm.001}$	.750$_{\pm.002}$	.750$_{\pm.001}$	.756$_{\pm.001}$	.748$_{\pm.002}$	**.769$_{\pm.001}$**	.768$_{\pm.002}$	.767$_{\pm.003}$	.765$_{\pm.001}$	.760$_{\pm.002}$
$A_{\text{comp}} \uparrow$	gpt2	.835$_{\pm.002}$	.833$_{\pm.004}$	.840$_{\pm.002}$	.841$_{\pm.001}$	.843$_{\pm.003}$	**.890$_{\pm.002}$**	.835$_{\pm.002}$	.887$_{\pm.002}$	.888$_{\pm.002}$	.888$_{\pm.001}$	.889$_{\pm.003}$	.840$_{\pm.003}$
	SmolLM-135M	.830$_{\pm.002}$	.835$_{\pm.001}$	.837$_{\pm.001}$	.837$_{\pm.002}$	.837$_{\pm.002}$	.851$_{\pm.001}$	.830$_{\pm.002}$	**.865$_{\pm.000}$**	**.865$_{\pm.001}$**	**.865$_{\pm.001}$**	**.865$_{\pm.002}$**	.846$_{\pm.002}$
$V_\ell \uparrow$	gpt2	.742	.966$_{\pm.003}$	**1.000$_{\pm.000}$**	**1.000$_{\pm.000}$**	**1.000$_{\pm.000}$**	.533$_{\pm.025}$	.523	.997$_{\pm.001}$	**1.000$_{\pm.000}$**	**1.000$_{\pm.000}$**	**1.000$_{\pm.000}$**	.536$_{\pm.015}$
	SmolLM-135M	.814	**1.000$_{\pm.000}$**	**1.000$_{\pm.000}$**	**1.000$_{\pm.000}$**	**1.000$_{\pm.000}$**	.587$_{\pm.019}$	.651	.982$_{\pm.001}$	.995$_{\pm.001}$	.996$_{\pm.000}$	.996$_{\pm.000}$	.494$_{\pm.006}$
$D_\ell \uparrow$	gpt2	.262	.178$_{\pm.002}$	.924$_{\pm.003}$	.974$_{\pm.002}$	.977$_{\pm.001}$	.990$_{\pm.001}$	.303	.111$_{\pm.003}$	.937$_{\pm.003}$	.977$_{\pm.001}$	.977$_{\pm.003}$	**.999$_{\pm.000}$**
	SmolLM-135M	.312	.517$_{\pm.003}$	.895$_{\pm.003}$	.925$_{\pm.000}$	.927$_{\pm.001}$	.960$_{\pm.002}$	.281	.642$_{\pm.005}$	.859$_{\pm.003}$	.862$_{\pm.002}$	.861$_{\pm.005}$	**.988$_{\pm.001}$**
bias $\downarrow$	gpt2	.0711	.1767$_{\pm.0050}$	.0080$_{\pm.0002}$	.0044$_{\pm.0002}$	.0040$_{\pm.0001}$	.0024$_{\pm.0002}$	.0599	.0971$_{\pm.0021}$	.0080$_{\pm.0002}$	.0043$_{\pm.0003}$	.0041$_{\pm.0004}$	**.0008$_{\pm.0001}$**
	SmolLM-135M	.0564	.0307$_{\pm.0001}$	.0100$_{\pm.0001}$	.0082$_{\pm.0001}$	.0080$_{\pm.0001}$	.0054$_{\pm.0001}$	.0583	.0242$_{\pm.0002}$	.0118$_{\pm.0002}$	.0115$_{\pm.0001}$	.0117$_{\pm.0004}$	**.0030$_{\pm.0001}$**
pref. $\to .5$	gpt2	.506	.729$_{\pm.003}$	.633$_{\pm.006}$	.540$_{\pm.009}$	**.499$_{\pm.011}$**	.512$_{\pm.013}$	.509	.716$_{\pm.005}$	.586$_{\pm.009}$	.476$_{\pm.016}$	.441$_{\pm.008}$	.483$_{\pm.048}$
	SmolLM-135M	.532	.666$_{\pm.005}$	.585$_{\pm.006}$	.529$_{\pm.010}$	.519$_{\pm.004}$	.489$_{\pm.014}$	**.490**	.611$_{\pm.002}$	.521$_{\pm.003}$	.480$_{\pm.002}$	.473$_{\pm.005}$	.511$_{\pm.003}$

Table 9: Generation in the unshuffling format on Blocks World at update 5,000. Each checkpoint produces two continuations for each of 200 held-out plans at temperature 1. Entries are mean fractions with 95% Student-t intervals across five training seeds. Valid and invalid-order outputs contain every chunk exactly once and terminate with EOS; invalid orders violate the DAG. All other outputs are malformed. Generation validity V_{gen} is the valid-output fraction. Diversity D_{gen} (distinct-valid/valid) divides the sum of distinct valid orders per plan by the total number of valid samples and is conditional on validity. Intermediate $k = 2, 4$ conditions cover GPT-2 and SmolLM-135M only. Bold compares k settings within each model: maxima for validity and diversity, minima for invalid-order and malformed rates, including ties.

model	k	V_{gen}	invalid-order	malformed	D_{gen}
GPT-2	1	**.998$_{\pm.002}$**	**.001$_{\pm.002}$**	**.002$_{\pm.003}$**	.542$_{\pm.013}$
	2	.990$_{\pm.006}$	.002$_{\pm.003}$	.007$_{\pm.004}$	.818$_{\pm.019}$
	4	.990$_{\pm.006}$	.003$_{\pm.004}$	.007$_{\pm.004}$	**.828$_{\pm.023}$**
	all	.989$_{\pm.009}$	.004$_{\pm.005}$	.007$_{\pm.005}$	**.828$_{\pm.021}$**
GPT-2-medium	1	**.999$_{\pm.003}$**	**.001$_{\pm.003}$**	**.000$_{\pm.000}$**	.520$_{\pm.008}$
	all	.992$_{\pm.004}$	**.001$_{\pm.003}$**	.007$_{\pm.004}$	**.837$_{\pm.012}$**
GPT-2-large	1	**1.000$_{\pm.001}$**	**.000$_{\pm.000}$**	**.001$_{\pm.001}$**	.512$_{\pm.004}$
	all	.990$_{\pm.002}$	.008$_{\pm.004}$	.002$_{\pm.003}$	**.838$_{\pm.008}$**
OPT-1.3B	1	**.995$_{\pm.004}$**	**.005$_{\pm.004}$**	.001$_{\pm.001}$	.533$_{\pm.011}$
	all	.958$_{\pm.047}$	.042$_{\pm.047}$	**.000$_{\pm.000}$**	**.829$_{\pm.015}$**
SmolLM-135M	1	.296$_{\pm.013}$	**.291$_{\pm.013}$**	.413$_{\pm.020}$	.875$_{\pm.021}$
	2	.341$_{\pm.025}$	.310$_{\pm.033}$	.350$_{\pm.045}$	.900$_{\pm.018}$
	4	.364$_{\pm.022}$	.314$_{\pm.042}$	**.322$_{\pm.057}$**	.892$_{\pm.011}$
	all	**.368$_{\pm.020}$**	.307$_{\pm.031}$	.324$_{\pm.033}$	**.920$_{\pm.019}$**
SmolLM-360M	1	.664$_{\pm.038}$	.299$_{\pm.034}$	.036$_{\pm.009}$	.794$_{\pm.015}$
	all	**.783$_{\pm.030}$**	**.186$_{\pm.021}$**	**.031$_{\pm.013}$**	**.829$_{\pm.026}$**
SmolLM-1.7B	1	**.998$_{\pm.003}$**	.001$_{\pm.002}$	**.001$_{\pm.003}$**	.510$_{\pm.004}$
	all	.995$_{\pm.000}$	**.000$_{\pm.000}$**	.005$_{\pm.000}$	**.836$_{\pm.008}$**
Qwen3-1.7B-Base	1	**1.000$_{\pm.000}$**	**.000$_{\pm.000}$**	**.000$_{\pm.000}$**	.505$_{\pm.003}$
	all	.993$_{\pm.003}$	.002$_{\pm.006}$	.005$_{\pm.004}$	**.817$_{\pm.007}$**
Pythia-1.4B	1	**.996$_{\pm.003}$**	**.002$_{\pm.003}$**	**.003$_{\pm.003}$**	.573$_{\pm.033}$
	all	.995$_{\pm.001}$	**.002$_{\pm.003}$**	.004$_{\pm.003}$	**.853$_{\pm.007}$**

B.3 NATURAL PROCEDURES AND OUT-OF-DOMAIN EVALUATION

In-domain proScript results use five seeds for all nine models. Out-of-domain evaluation of proScript-trained models on ERFGC and PCS uses five seeds for GPT-2 and SmolLM-135M and seed-0 point estimates for the other seven models. ERFGC and PCS are evaluated through probes and likelihoods; sampled generation is evaluated in-domain on proScript. PCS uses the enumerable subset of derived graphs. Its non-reachability labels and entropy over valid orders inherit the uncertainty of that graph construction.

Table 10: In-domain proScript results at update 2,000 for all nine models and $k \in \{1, \text{all}\}$. All trained settings use five seeds; columns and metrics follow Appendix B.

		LM				Unshuffling			
	model	base	$k=1$	$k=$ all	Control	base	$k=1$	$k=$ all	Control
$A_{\text{inc}} \uparrow$	gpt2	.569±.011	.567±.008	.574±.008	.583±.019	.569±.011	**.611±.020**	.600±.014	.598±.024
	gpt2-medium	.577±.017	.570±.016	.574±.016	.568±.024	.577±.017	**.593±.015**	**.593±.023**	.586±.021
	gpt2-large	.569±.014	.554±.014	.565±.010	.570±.014	.569±.014	**.580±.015**	.579±.012	.572±.011
	opt-1.3b	.552±.020	.542±.034	.552±.009	.553±.010	.552±.020	.554±.018	**.557±.010**	.545±.011
	SmolLM-135M	.580±.015	.583±.009	.584±.010	.579±.016	.580±.015	.584±.014	**.588±.008**	.583±.014
	SmolLM-360M	.588±.011	.575±.018	.569±.021	.584±.013	.588±.011	**.590±.016**	.586±.011	.575±.012
	SmolLM-1.7B	.542±.016	.539±.019	.557±.018	.546±.013	.542±.016	.563±.015	**.564±.019**	.551±.012
	pythia-1.4b	.543±.017	.541±.015	.545±.010	.546±.011	.543±.017	.549±.008	**.555±.015**	.541±.015
	Qwen3-1.7B-Base	.550±.028	.534±.017	.539±.009	.540±.019	.550±.028	.543±.015	**.556±.016**	.550±.009
$A_{\text{comp}} \uparrow$	gpt2	.662±.003	.656±.007	.660±.006	.661±.007	.662±.003	**.678±.007**	.676±.007	.655±.006
	gpt2-medium	.671±.004	.667±.006	.668±.004	.657±.004	.671±.004	.690±.008	**.691±.005**	.663±.003
	gpt2-large	.660±.007	.661±.005	.664±.005	.655±.006	.660±.007	**.687±.005**	.685±.004	.653±.004
	opt-1.3b	.649±.005	.647±.006	.640±.011	.636±.008	.649±.005	.675±.005	**.678±.006**	.633±.008
	SmolLM-135M	.663±.003	.675±.006	.675±.006	**.684±.005**	.663±.003	.679±.007	.680±.008	.667±.005
	SmolLM-360M	.664±.003	.674±.005	.674±.004	**.678±.003**	.664±.003	.672±.005	.672±.002	.664±.006
	SmolLM-1.7B	.641±.008	.624±.009	.635±.007	.623±.006	.641±.008	.677±.004	**.686±.011**	.636±.005
	pythia-1.4b	.635±.007	.626±.005	.634±.007	.639±.006	.635±.007	.656±.008	**.660±.012**	.627±.002
	Qwen3-1.7B-Base	.629±.003	.638±.005	.639±.009	.633±.005	.629±.003	.689±.011	**.703±.007**	.626±.006
$V_\ell \uparrow$	gpt2	.689	.820±.005	**.821±.004**	.542±.005	.523	.774±.006	.783±.004	.492±.012
	gpt2-medium	.770	**.824±.006**	.817±.003	.529±.008	.523	.799±.002	.805±.002	.504±.006
	gpt2-large	.775	**.808±.013**	.807±.003	.535±.012	.552	.797±.006	.801±.003	.503±.013
	opt-1.3b	.799	.804±.007	.791±.010	.532±.009	.590	.814±.006	**.821±.012**	.504±.012
	SmolLM-135M	.727	**.867±.002**	.864±.002	.637±.006	.545	.698±.005	.696±.005	.492±.002
	SmolLM-360M	.752	**.880±.002**	.879±.002	.624±.006	.540	.745±.002	.747±.002	.486±.008
	SmolLM-1.7B	.793	.827±.008	.816±.004	.553±.009	.586	.842±.010	**.854±.007**	.507±.009
	pythia-1.4b	.786	**.797±.013**	.789±.006	.536±.009	.553	.786±.007	.796±.009	.505±.021
	Qwen3-1.7B-Base	.758	.814±.002	.805±.004	.544±.005	.581	.896±.000	**.912±.005**	.496±.006
$D_\ell \uparrow$	gpt2	.656	.242±.016	.299±.015	.595±.012	.343	.282±.010	.456±.014	**.886±.005**
	gpt2-medium	.641	.150±.023	.237±.020	.479±.019	.390	.192±.021	.442±.019	**.910±.003**
	gpt2-large	.588	.173±.016	.278±.020	.517±.019	.327	.206±.028	.464±.009	**.910±.012**
	opt-1.3b	.583	.168±.035	.276±.030	.508±.028	.347	.180±.016	.476±.024	**.910±.009**
	SmolLM-135M	.574	.528±.010	.551±.001	.719±.009	.393	.730±.003	.742±.004	**.857±.004**
	SmolLM-360M	.593	.546±.007	.558±.007	.750±.007	.460	.732±.003	.753±.003	**.889±.003**
	SmolLM-1.7B	.549	.119±.017	.181±.010	.386±.019	.566	.195±.027	.495±.010	**.923±.003**
	pythia-1.4b	.571	.160±.003	.264±.015	.486±.017	.466	.203±.018	.452±.035	**.872±.017**
	Qwen3-1.7B-Base	.578	.106±.011	.212±.031	.403±.026	.722	.173±.015	.556±.015	**.945±.003**
bias $\downarrow$	gpt2	.0379	.1377±.0032	.1115±.0025	.0477±.0012	.0914	.1101±.0020	.0716±.0005	**.0143±.0004**
	gpt2-medium	.0418	.2217±.0063	.1619±.0058	.0680±.0023	.0840	.1912±.0061	.0823±.0028	**.0120±.0005**
	gpt2-large	.0454	.2013±.0146	.1381±.0040	.0582±.0033	.1119	.1954±.0083	.0806±.0012	**.0121±.0006**
	opt-1.3b	.0489	.2197±.0092	.1403±.0037	.0624±.0061	.0835	.1979±.0050	.0771±.0044	**.0132±.0014**
	SmolLM-135M	.0457	.0558±.0006	.0532±.0004	.0310±.0004	.0780	.0307±.0003	.0291±.0004	**.0182±.0001**
	SmolLM-360M	.0449	.0603±.0004	.0572±.0005	.0299±.0002	.0661	.0327±.0003	.0305±.0005	**.0152±.0001**
	SmolLM-1.7B	.0548	.3295±.0150	.2147±.0089	.0851±.0032	.0549	.2222±.0087	.0821±.0019	**.0101±.0001**
	pythia-1.4b	.0456	.2180±.0048	.1473±.0075	.0641±.0045	.0632	.1997±.0115	.0792±.0092	**.0161±.0022**
	Qwen3-1.7B-Base	.0511	.3143±.0156	.2045±.0042	.0917±.0044	.0307	.2632±.0089	.0725±.0047	**.0084±.0003**
pref. $\rightarrow .5$	gpt2	**.510**	.527±.024	.527±.009	.528±.011	.515	.480±.015	.484±.018	.469±.009
	gpt2-medium	**.508**	.524±.025	.530±.014	.514±.015	**.508**	.475±.011	.481±.026	.528±.029
	gpt2-large	.523	.493±.020	**.501±.011**	.487±.026	.525	.486±.017	.491±.026	.521±.016
	opt-1.3b	.513	.482±.024	.514±.010	.505±.033	**.503**	.487±.034	.488±.014	.512±.028
	SmolLM-135M	**.510**	.484±.009	.484±.011	.544±.012	.472	.474±.008	.489±.007	.473±.013
	SmolLM-360M	.510	.507±.007	**.497±.004**	**.503±.019**	**.503**	.488±.006	.483±.005	.467±.024
	SmolLM-1.7B	.490	.515±.017	**.502±.023**	.494±.021	.482	.504±.008	.491±.022	.507±.020
	pythia-1.4b	.487	**.496±.022**	.511±.024	.510±.033	.487	.517±.032	.483±.010	.531±.025
	Qwen3-1.7B-Base	.490	**.500±.030**	.476±.016	.524±.034	.477	.515±.017	.537±.027	.551±.032

Table 11: In-domain proScript results at update 2,000 for GPT-2 and SmolLM-135M with $k \in \{1, 2, 4, \text{all}\}$. All trained settings use five seeds; columns and metrics follow Appendix B.

		LM						Unshuffling					
	model	base	$k=1$	$k=2$	$k=4$	$k=\text{all}$	Control	base	$k=1$	$k=2$	$k=4$	$k=\text{all}$	Control
$A_{\text{inc}} \uparrow$	gpt2	.569±.011	.567±.008	.571±.014	.570±.012	.574±.008	.583±.019	.569±.011	**.611±.020**	.602±.014	.598±.021	.600±.014	.598±.024
	SmolLM-135M	.580±.015	.583±.009	.583±.004	.585±.010	.584±.010	.579±.016	.580±.015	.584±.014	.587±.004	.585±.009	**.588±.008**	.583±.014
$A_{\text{comp}} \uparrow$	gpt2	.662±.003	.656±.007	.658±.005	.659±.007	.660±.006	.661±.007	.662±.003	**.678±.007**	.676±.005	.677±.006	.676±.007	.655±.006
	SmolLM-135M	.663±.003	.675±.006	.675±.007	.675±.007	.675±.006	**.684±.005**	.663±.003	.679±.007	.679±.008	.679±.009	.680±.008	.667±.005
$V_\ell \uparrow$	gpt2	.689	.820±.005	.820±.007	**.822±.005**	.821±.004	.542±.005	.523	.774±.006	.781±.002	.784±.004	.783±.004	.492±.012
	SmolLM-135M	.727	**.867±.002**	.864±.002	.864±.002	.864±.002	.637±.006	.545	.698±.005	.695±.003	.696±.004	.696±.005	.492±.002
$D_\ell \uparrow$	gpt2	.656	.242±.016	.293±.013	.292±.013	.299±.015	.595±.012	.343	.282±.010	.425±.025	.455±.010	.456±.014	**.886±.005**
	SmolLM-135M	.574	.528±.010	.546±.004	.548±.005	.551±.001	.719±.009	.393	.730±.003	.737±.004	.739±.003	.742±.004	**.857±.004**
bias $\downarrow$	gpt2	.0379	.1377±.0032	.1174±.0018	.1136±.0032	.1115±.0025	.0477±.0012	.0914	.1101±.0020	.0803±.0026	.0726±.0017	.0716±.0005	**.0143±.0004**
	SmolLM-135M	.0457	.0558±.0006	.0542±.0004	.0536±.0006	.0532±.0004	.0310±.0004	.0780	.0307±.0003	.0293±.0004	.0290±.0005	.0291±.0004	**.0182±.0001**
pref. $\to .5$	gpt2	**.510**	.527±.024	.530±.012	.526±.011	.527±.009	.528±.011	.515	.480±.015	.476±.016	.474±.014	.484±.018	.469±.009
	SmolLM-135M	**.510**	.484±.009	.486±.007	.488±.018	.484±.011	.544±.012	.472	.474±.008	.482±.009	.484±.009	.489±.007	.473±.013

Table 12: Generation in the unshuffling format on proScript at update 2,000. Each checkpoint produces two continuations for each of 200 held-out plans at temperature 1. Entries are mean fractions with 95% Student-t intervals across five training seeds. Invalid-order outputs are exact chunk permutations that terminate with EOS but violate the DAG; all remaining non-valid outputs are malformed. Validity V_{gen} and diversity D_{gen} (distinct-valid/valid) are defined as in Table 9. Intermediate $k = 2, 4$ conditions cover GPT-2 and SmolLM-135M only. Bold follows the same rule as Table 9.

model	k	V_{gen}	invalid-order	malformed	D_{gen}
GPT-2	1	.067±.008	.634±.034	**.299±.036**	.692±.054
	2	**.068±.018**	.624±.034	.308±.034	.676±.047
	4	.059±.010	.616±.015	.325±.012	.701±.057
	all	.067±.008	**.604±.019**	.330±.013	**.745±.032**
GPT-2-medium	1	.097±.014	.725±.024	.178±.019	.630±.027
	all	**.105±.017**	**.721±.031**	**.175±.020**	**.656±.037**
GPT-2-large	1	**.098±.011**	**.685±.031**	**.216±.028**	.620±.046
	all	.094±.016	.689±.035	.217±.027	**.692±.038**
OPT-1.3B	1	.118±.038	**.480±.026**	.401±.044	.586±.041
	all	**.123±.029**	.556±.067	**.321±.091**	**.668±.039**
SmolLM-135M	1	**.009±.003**	.344±.014	**.647±.013**	**1.000±.000**
	2	.008±.004	.340±.017	.652±.018	.950±.139
	4	.007±.003	.336±.012	.657±.013	.900±.278
	all	.005±.003	**.332±.028**	.663±.030	.933±.185
SmolLM-360M	1	**.015±.009**	**.575±.022**	.410±.024	.913±.107
	all	.014±.005	.584±.028	**.403±.032**	**1.000±.000**
SmolLM-1.7B	1	.137±.025	**.669±.031**	.194±.026	.588±.045
	all	**.167±.015**	.703±.007	**.131±.016**	**.616±.024**
Qwen3-1.7B-Base	1	.337±.022	.610±.026	**.054±.017**	.574±.025
	all	**.370±.010**	**.569±.019**	.061±.022	**.634±.019**
Pythia-1.4B	1	.062±.009	**.385±.032**	.552±.026	.577±.046
	all	**.066±.023**	.462±.104	**.472±.127**	**.667±.069**

Table 13: ProScript checkpoints evaluated on ERFGC at update 2,000: all nine models and $k \in \{1, \text{all}\}$. GPT-2 and SmolLM-135M use five seeds; other models are seed-0 point estimates. Columns and metrics follow Appendix B.

		LM				Unshuffling			
	model	base	$k=1$	$k=\text{all}$	Control	base	$k=1$	$k=\text{all}$	Control
$A_{\text{inc}} \uparrow$	gpt2	**.745**	.739$_{\pm.033}$	.732$_{\pm.040}$	.741$_{\pm.010}$	**.745**	.712$_{\pm.023}$	.729$_{\pm.027}$	.706$_{\pm.035}$
	gpt2-medium	**.782**	.726	.714	.730	**.782**	.716	.714	.736
	gpt2-large	**.753**	.749	.710	.737	**.753**	.726	.712	.701
	opt-1.3b	**.799**	.790	.761	.747	**.799**	.743	.778	.755
	SmolLM-135M	.712	.713$_{\pm.018}$	.714$_{\pm.016}$	**.724$_{\pm.016}$**	.712	.707$_{\pm.031}$	.703$_{\pm.022}$	.692$_{\pm.029}$
	SmolLM-360M	**.790**	.755	.764	.768	**.790**	.766	.757	.745
	SmolLM-1.7B	**.809**	.793	.795	.778	**.809**	.774	.784	.707
	pythia-1.4b	**.778**	.763	.724	.736	**.778**	.722	.747	.678
	Qwen3-1.7B-Base	**.809**	.774	.753	.772	**.809**	.774	.774	.714
$A_{\text{comp}} \uparrow$	gpt2	.768	.762$_{\pm.022}$	.763$_{\pm.020}$	**.774$_{\pm.006}$**	.768	.734$_{\pm.012}$	.732$_{\pm.012}$	.719$_{\pm.018}$
	gpt2-medium	**.816**	.793	.782	.799	**.816**	.760	.769	.764
	gpt2-large	.810	.799	.782	**.813**	.810	.760	.748	.730
	opt-1.3b	**.847**	.805	.802	.802	**.847**	.782	.805	.765
	SmolLM-135M	**.817**	.790$_{\pm.015}$	.789$_{\pm.014}$	.787$_{\pm.014}$	**.817**	.775$_{\pm.018}$	.769$_{\pm.018}$	.739$_{\pm.010}$
	SmolLM-360M	.799	**.804**	**.804**	.802	.799	.793	.793	.770
	SmolLM-1.7B	.837	.829	.839	**.843**	.837	.828	.838	.763
	pythia-1.4b	**.834**	.794	.805	.770	**.834**	.775	.767	.761
	Qwen3-1.7B-Base	**.845**	.829	.816	.819	**.845**	.835	.816	.740
$V_\ell \uparrow$	gpt2	**.834**	.645$_{\pm.017}$	.652$_{\pm.014}$	.599$_{\pm.024}$	.477	.503$_{\pm.010}$	.512$_{\pm.009}$	.469$_{\pm.008}$
	gpt2-medium	**.885**	.701	.689	.599	.503	.537	.523	.486
	gpt2-large	**.896**	.645	.664	.560	.489	.536	.547	.479
	opt-1.3b	**.926**	.600	.585	.483	.514	.571	.622	.531
	SmolLM-135M	**.856**	.825$_{\pm.004}$	.827$_{\pm.006}$	.813$_{\pm.006}$	.487	.517$_{\pm.005}$	.512$_{\pm.006}$	.468$_{\pm.006}$
	SmolLM-360M	.886	.881	**.889**	.859	.493	.521	.519	.456
	SmolLM-1.7B	**.915**	.762	.773	.638	.528	.662	.653	.502
	pythia-1.4b	**.936**	.614	.598	.515	.541	.543	.545	.501
	Qwen3-1.7B-Base	**.914**	.736	.707	.592	.582	.677	.713	.528
$D_\ell \uparrow$	gpt2	.271	.065$_{\pm.012}$	.074$_{\pm.006}$	.115$_{\pm.018}$	.139	.208$_{\pm.014}$	.253$_{\pm.026}$	**.491$_{\pm.016}$**
	gpt2-medium	.291	.043	.038	.093	.132	.135	.240	**.527**
	gpt2-large	.197	.053	.071	.131	.122	.183	.212	**.550**
	opt-1.3b	.202	.096	.095	.104	.130	.172	.267	**.442**
	SmolLM-135M	.156	.202$_{\pm.009}$	.198$_{\pm.005}$	.218$_{\pm.007}$	.139	.262$_{\pm.011}$	.263$_{\pm.007}$	**.333$_{\pm.008}$**
	SmolLM-360M	.161	.178	.167	.179	.165	.277	.279	**.462**
	SmolLM-1.7B	.160	.053	.039	.094	.170	.192	.215	**.565**
	pythia-1.4b	.145	.078	.095	.136	.163	.109	.199	**.396**
	Qwen3-1.7B-Base	.219	.026	.069	.080	.297	.142	.205	**.484**
bias $\downarrow$	gpt2	.0276	.1154$_{\pm.0065}$	.1164$_{\pm.0084}$	.0742$_{\pm.0026}$	.0458	.0374$_{\pm.0011}$	.0312$_{\pm.0011}$	**.0148$_{\pm.0001}$**
	gpt2-medium	.0297	.1339	.1548	.0977	.0439	.0566	.0343	**.0148**
	gpt2-large	.0355	.1128	.1222	.0699	.0483	.0462	.0362	**.0150**
	opt-1.3b	.0351	.1079	.1083	.0990	.0433	.0577	.0338	**.0214**
	SmolLM-135M	.0412	.0402$_{\pm.0006}$	.0398$_{\pm.0003}$	.0324$_{\pm.0003}$	.0431	.0270$_{\pm.0003}$	.0266$_{\pm.0003}$	**.0201$_{\pm.0003}$**
	SmolLM-360M	.0438	.0431	.0427	.0335	.0403	.0254	.0248	**.0149**
	SmolLM-1.7B	.0420	.1490	.1424	.0946	.0365	.0453	.0354	**.0121**
	pythia-1.4b	.0421	.0840	.0875	.0648	.0387	.0622	.0421	**.0222**
	Qwen3-1.7B-Base	.0367	.1599	.1488	.0816	.0216	.0563	.0384	**.0168**
pref. $\to .5$	gpt2	.615	.451$_{\pm.016}$	.450$_{\pm.013}$	.467$_{\pm.014}$	.478	.487$_{\pm.019}$	.490$_{\pm.022}$	**.502$_{\pm.011}$**
	gpt2-medium	.596	.491	.484	.520	.534	.544	**.503**	.536
	gpt2-large	.629	.528	**.505**	.516	.540	.528	.532	.464
	opt-1.3b	.634	.482	**.503**	.507	.551	.524	.534	.480
	SmolLM-135M	.557	.585$_{\pm.007}$	.586$_{\pm.006}$	.578$_{\pm.011}$	**.505**	.552$_{\pm.009}$	.550$_{\pm.005}$	.527$_{\pm.014}$
	SmolLM-360M	.547	.584	.571	.574	.538	.563	.561	**.532**
	SmolLM-1.7B	.640	.574	.586	.484	.530	.528	.532	**.495**
	pythia-1.4b	.627	.484	.518	.544	.557	.528	.540	**.503**
	Qwen3-1.7B-Base	.588	.578	.545	.538	**.487**	.526	.580	.456

Table 14: ProScript checkpoints evaluated on ERFGC at update 2,000: GPT-2 and SmolLM-135M with $k \in \{1, 2, 4, \text{all}\}$. Both models use five seeds; columns and metrics follow Appendix B.

		LM						Unshuffling					
	model	base	$k=1$	$k=2$	$k=4$	$k=\text{all}$	Control	base	$k=1$	$k=2$	$k=4$	$k=\text{all}$	Control
$A_{\text{inc}} \uparrow$	gpt2	**.745**	.739$_{\pm.033}$	.736$_{\pm.046}$	.734$_{\pm.039}$	.732$_{\pm.040}$	.741$_{\pm.010}$	**.745**	.712$_{\pm.023}$	.712$_{\pm.025}$	.721$_{\pm.026}$	.729$_{\pm.027}$	.706$_{\pm.035}$
	SmolLM-135M	.712	.713$_{\pm.018}$	.718$_{\pm.016}$	.716$_{\pm.017}$	.714$_{\pm.016}$	**.724$_{\pm.016}$**	.712	.707$_{\pm.031}$	.700$_{\pm.030}$	.701$_{\pm.031}$	.703$_{\pm.022}$	.692$_{\pm.029}$
$A_{\text{comp}} \uparrow$	gpt2	.768	.762$_{\pm.022}$	.762$_{\pm.017}$	.763$_{\pm.019}$	.763$_{\pm.020}$	**.774$_{\pm.006}$**	.768	.734$_{\pm.012}$	.732$_{\pm.012}$	.730$_{\pm.010}$	.732$_{\pm.012}$	.719$_{\pm.018}$
	SmolLM-135M	**.817**	.790$_{\pm.015}$	.790$_{\pm.014}$	.790$_{\pm.014}$	.789$_{\pm.014}$	.787$_{\pm.014}$	**.817**	.775$_{\pm.018}$	.770$_{\pm.022}$	.771$_{\pm.014}$	.769$_{\pm.018}$	.739$_{\pm.010}$
$V_\ell \uparrow$	gpt2	**.834**	.645$_{\pm.017}$	.646$_{\pm.012}$	.651$_{\pm.006}$	.652$_{\pm.014}$	.599$_{\pm.024}$	.477	.503$_{\pm.010}$	.511$_{\pm.020}$	.516$_{\pm.005}$	.512$_{\pm.009}$	.469$_{\pm.008}$
	SmolLM-135M	**.856**	.825$_{\pm.004}$	.825$_{\pm.005}$	.824$_{\pm.005}$	.827$_{\pm.006}$	.813$_{\pm.006}$	.487	.517$_{\pm.005}$	.516$_{\pm.004}$	.514$_{\pm.006}$	.512$_{\pm.006}$	.468$_{\pm.006}$
$D_\ell \uparrow$	gpt2	.271	.065$_{\pm.012}$	.079$_{\pm.008}$	.075$_{\pm.010}$	.074$_{\pm.006}$	.115$_{\pm.018}$	.139	.208$_{\pm.014}$	.239$_{\pm.027}$	.233$_{\pm.016}$	.253$_{\pm.026}$	**.491$_{\pm.016}$**
	SmolLM-135M	.156	.202$_{\pm.009}$	.201$_{\pm.009}$	.198$_{\pm.007}$	.198$_{\pm.005}$	.218$_{\pm.007}$	.139	.262$_{\pm.011}$	.261$_{\pm.007}$	.262$_{\pm.008}$	.263$_{\pm.007}$	**.333$_{\pm.008}$**
bias $\downarrow$	gpt2	.0276	.1154$_{\pm.0065}$	.1131$_{\pm.0075}$	.1142$_{\pm.0049}$	.1164$_{\pm.0084}$	.0742$_{\pm.0026}$	.0458	.0374$_{\pm.0011}$	.0333$_{\pm.0009}$	.0323$_{\pm.0013}$	.0312$_{\pm.0011}$	**.0148$_{\pm.0001}$**
	SmolLM-135M	.0412	.0402$_{\pm.0006}$	.0398$_{\pm.0005}$	.0397$_{\pm.0007}$	.0398$_{\pm.0003}$	.0324$_{\pm.0003}$	.0431	.0270$_{\pm.0003}$	.0270$_{\pm.0001}$	.0269$_{\pm.0004}$	.0266$_{\pm.0003}$	**.0201$_{\pm.0003}$**
pref. $\to .5$	gpt2	.615	.451$_{\pm.016}$	.458$_{\pm.011}$	.453$_{\pm.010}$	.450$_{\pm.013}$	.467$_{\pm.014}$	.478	.487$_{\pm.019}$	.484$_{\pm.031}$	.495$_{\pm.024}$	.490$_{\pm.022}$	**.502$_{\pm.011}$**
	SmolLM-135M	.557	.585$_{\pm.007}$	.587$_{\pm.009}$	.589$_{\pm.005}$	.586$_{\pm.006}$	.578$_{\pm.011}$	**.505**	.552$_{\pm.009}$	.551$_{\pm.005}$	.549$_{\pm.011}$	.550$_{\pm.005}$	.527$_{\pm.014}$

Table 15: ProScript checkpoints evaluated on PCS at update 2,000: all nine models and $k \in \{1, \text{all}\}$. GPT-2 and SmolLM-135M use five seeds; other models are seed-0 point estimates. The evaluation graphs are derived rather than annotated; columns and metrics follow Appendix B.

		LM				Unshuffling			
	model	base	$k=1$	$k=\text{all}$	Control	base	$k=1$	$k=\text{all}$	Control
$A_{\text{inc}} \uparrow$	gpt2	**.743**	.723$_{\pm.010}$	.725$_{\pm.006}$	.719$_{\pm.006}$	**.743**	.729$_{\pm.007}$	.732$_{\pm.007}$	.725$_{\pm.010}$
	gpt2-medium	.727	.722	.721	.729	.727	**.737**	.732	.734
	gpt2-large	.719	.727	.717	**.747**	.719	.728	.727	.722
	opt-1.3b	.751	.734	.747	.750	.751	**.753**	.752	.751
	SmolLM-135M	**.736**	.728$_{\pm.009}$	.729$_{\pm.009}$	.730$_{\pm.009}$	**.736**	.735$_{\pm.016}$	.734$_{\pm.014}$	.731$_{\pm.015}$
	SmolLM-360M	.736	.738	.738	.741	.736	.753	**.756**	.726
	SmolLM-1.7B	.750	**.756**	.753	.754	.750	.750	.746	.727
	pythia-1.4b	.734	.753	.747	**.754**	.734	.732	.741	.726
	Qwen3-1.7B-Base	**.759**	.743	.743	.747	**.759**	.752	.754	.726
$A_{\text{comp}} \uparrow$	gpt2	.657	.652$_{\pm.008}$	.650$_{\pm.005}$	**.665$_{\pm.003}$**	.657	.649$_{\pm.014}$	.651$_{\pm.012}$	.653$_{\pm.006}$
	gpt2-medium	**.659**	.642	.639	.644	**.659**	.637	.640	.652
	gpt2-large	**.633**	.612	.618	.621	**.633**	.611	.623	.624
	opt-1.3b	**.632**	.598	.616	.620	**.632**	.619	.617	.608
	SmolLM-135M	.669	**.675$_{\pm.006}$**	.674$_{\pm.006}$	**.675$_{\pm.004}$**	.669	.662$_{\pm.007}$	.662$_{\pm.005}$	.654$_{\pm.007}$
	SmolLM-360M	.647	.645	.649	**.651**	.647	.640	.644	.646
	SmolLM-1.7B	.627	.613	.630	**.637**	.627	.633	.629	.619
	pythia-1.4b	**.636**	.616	.625	.622	**.636**	.607	.629	.612
	Qwen3-1.7B-Base	**.632**	.622	.626	.612	**.632**	.615	.631	.613
$V_\ell \uparrow$	gpt2	**.674**	.618$_{\pm.006}$	.617$_{\pm.003}$	.571$_{\pm.005}$	.554	.530$_{\pm.005}$	.531$_{\pm.004}$	.516$_{\pm.005}$
	gpt2-medium	**.738**	.636	.650	.590	.564	.534	.531	.506
	gpt2-large	**.740**	.624	.636	.575	.572	.530	.536	.500
	opt-1.3b	**.764**	.588	.581	.524	.583	.526	.527	.475
	SmolLM-135M	.686	**.691$_{\pm.004}$**	.690$_{\pm.001}$	.659$_{\pm.003}$	.568	.525$_{\pm.002}$	.522$_{\pm.003}$	.504$_{\pm.002}$
	SmolLM-360M	**.727**	.717	.712	.681	.571	.549	.551	.515
	SmolLM-1.7B	**.783**	.664	.657	.593	.605	.599	.569	.502
	pythia-1.4b	**.755**	.577	.587	.533	.594	.519	.527	.492
	Qwen3-1.7B-Base	**.773**	.645	.643	.581	.601	.605	.599	.504
$D_\ell \uparrow$	gpt2	.332	.099$_{\pm.005}$	.103$_{\pm.006}$	.170$_{\pm.007}$	.146	.152$_{\pm.008}$	.187$_{\pm.004}$	**.368$_{\pm.004}$**
	gpt2-medium	.278	.077	.082	.132	.173	.111	.190	**.398**
	gpt2-large	.266	.099	.103	.168	.137	.128	.176	**.372**
	opt-1.3b	.213	.102	.113	.150	.157	.135	.200	**.352**
	SmolLM-135M	.282	.238$_{\pm.001}$	.243$_{\pm.003}$	.297$_{\pm.002}$	.166	.293$_{\pm.002}$	.299$_{\pm.003}$	**.344$_{\pm.004}$**
	SmolLM-360M	.231	.196	.197	.267	.171	.313	.314	**.418**
	SmolLM-1.7B	.195	.062	.068	.121	.182	.121	.177	**.403**
	pythia-1.4b	.175	.101	.123	.168	.192	.107	.164	**.321**
	Qwen3-1.7B-Base	.188	.054	.063	.131	.301	.110	.177	**.367**
bias $\downarrow$	gpt2	**.0391**	.1325$_{\pm.0019}$	.1287$_{\pm.0034}$	.1034$_{\pm.0006}$	.0669	.1194$_{\pm.0028}$	.1020$_{\pm.0043}$	.0858$_{\pm.0035}$
	gpt2-medium	**.0426**	.1634	.1572	.1194	.0618	.1366	.0932	.0815
	gpt2-large	**.0443**	.1228	.1212	.0951	.0701	.1228	.1020	.1178
	opt-1.3b	**.0469**	.1228	.1195	.1001	.0613	.1253	.0997	.1104
	SmolLM-135M	**.0420**	.0479$_{\pm.0001}$	.0472$_{\pm.0002}$	.0424$_{\pm.0001}$	.0528	.0487$_{\pm.0003}$	.0484$_{\pm.0003}$	.0506$_{\pm.0003}$
	SmolLM-360M	.0472	.0569	.0557	.0467	.0531	.0493	.0487	**.0441**
	SmolLM-1.7B	.0498	.1803	.1625	.1268	**.0479**	.1202	.0910	.1003
	pythia-1.4b	.0547	.1267	.1105	.0924	**.0529**	.1780	.1140	.1207
	Qwen3-1.7B-Base	.0559	.1986	.1734	.1200	**.0384**	.1448	.1020	.1485
pref. $\to .5$	gpt2	.525	.515$_{\pm.004}$	.518$_{\pm.003}$	.517$_{\pm.006}$	.513	.497$_{\pm.006}$	**.499$_{\pm.003}$**	.487$_{\pm.002}$
	gpt2-medium	.530	.493	.487	.505	.527	**.498**	.508	.486
	gpt2-large	**.501**	.484	.485	**.501**	.517	.497	**.501**	.492
	opt-1.3b	.515	**.500**	.497	.513	.505	.498	.494	.489
	SmolLM-135M	**.500**	.499$_{\pm.002}$	.496$_{\pm.001}$	.511$_{\pm.001}$	.498	.514$_{\pm.001}$	.513$_{\pm.002}$	.510$_{\pm.002}$
	SmolLM-360M	.513	.511	.513	.521	.512	**.500**	.496	.499
	SmolLM-1.7B	.510	.489	.502	.502	**.499**	.473	.510	.503
	pythia-1.4b	.519	.479	.496	.516	.510	.504	.518	**.498**
	Qwen3-1.7B-Base	.513	.486	.493	.495	**.504**	.484	.518	.509

Table 16: ProScript checkpoints evaluated on PCS at update 2,000: GPT-2 and SmolLM-135M with $k \in \{1, 2, 4, \text{all}\}$. Both models use five seeds. The evaluation graphs are derived; columns and metrics follow Appendix B.

		LM						Unshuffling					
	model	base	$k=1$	$k=2$	$k=4$	$k=\text{all}$	Control	base	$k=1$	$k=2$	$k=4$	$k=\text{all}$	Control
$A_{\text{inc}} \uparrow$	gpt2	**.743**	.723±.010	.723±.004	.722±.006	.725±.006	.719±.006	**.743**	.729±.007	.734±.005	.731±.010	.732±.007	.725±.010
	SmolLM-135M	**.736**	.728±.009	.729±.008	.728±.008	.729±.009	.730±.009	**.736**	.735±.016	.735±.014	.735±.014	.734±.014	.731±.015
$A_{\text{comp}} \uparrow$	gpt2	.657	.652±.008	.654±.007	.652±.006	.650±.005	**.665±.003**	.657	.649±.014	.647±.016	.649±.010	.651±.012	.653±.006
	SmolLM-135M	.669	**.675±.006**	**.675±.006**	.674±.005	.674±.006	**.675±.004**	.669	.662±.007	.663±.008	.662±.008	.662±.005	.654±.007
$V_\ell \uparrow$	gpt2	**.674**	.618±.006	.618±.005	.616±.008	.617±.003	.571±.005	.554	.530±.005	.532±.004	.532±.008	.531±.004	.516±.005
	SmolLM-135M	.686	**.691±.004**	.687±.002	.690±.002	.690±.001	.659±.003	.568	.525±.002	.527±.005	.523±.002	.522±.003	.504±.002
$D_\ell \uparrow$	gpt2	.332	.099±.005	.103±.004	.101±.005	.103±.006	.170±.007	.146	.152±.008	.175±.009	.184±.006	.187±.004	**.368±.004**
	SmolLM-135M	.282	.238±.001	.242±.003	.242±.001	.243±.003	.297±.002	.166	.293±.002	.296±.006	.299±.004	.299±.003	**.344±.004**
bias $\downarrow$	gpt2	**.0391**	.1325±.0019	.1298±.0028	.1288±.0028	.1287±.0034	.1034±.0006	.0669	.1194±.0028	.1077±.0028	.1038±.0028	.1020±.0043	.0858±.0035
	SmolLM-135M	**.0420**	.0479±.0001	.0474±.0001	.0474±.0001	.0472±.0002	.0424±.0001	.0528	.0487±.0003	.0486±.0005	.0483±.0005	.0484±.0003	.0506±.0003
pref. $\to .5$	gpt2	.525	.515±.004	.518±.004	.518±.005	.518±.003	.517±.006	.513	.497±.006	**.500±.004**	**.500±.003**	.499±.003	.487±.002
	SmolLM-135M	**.500**	.499±.002	.497±.003	.498±.002	.496±.001	.511±.001	.498	.514±.001	.513±.001	.513±.002	.513±.002	.510±.002

B.4 Out-of-domain evaluation after RecipeNLG training

These results use the available seed-0 RecipeNLG checkpoints at update 45,000. Each training recipe supplies one observed target order, with no annotated partial order. We compare LM and unshuffling at $k = 1$ with their invalid-target controls. Evaluations on ERFGC, PCS, and proScript cover all nine models. These point estimates support descriptive comparisons without training-seed confidence intervals or significance claims.

Table 17: RecipeNLG checkpoints evaluated on ERFGC, PCS, and proScript at update 45,000. All nine models, seed 0, with $k = 1$ denoting the observed recipe order. The PCS evaluation graphs are derived; columns and metrics follow Appendix B. Bold marks row maxima within each dataset, including controls and ties; for bias and pref., it marks minima and values closest to .5, respectively.

		ERFGC						PCS						proScript					
		LM			Unshuffling			LM			Unshuffling			LM			Unshuffling		
	model	base	$k=1$	Control	base	$k=1$	Control	base	$k=1$	Control	base	$k=1$	Control	base	$k=1$	Control	base	$k=1$	Control
$A_{\text{inc}} \uparrow$	gpt2	.745	.759	.768	.745	**.795**	.755	.743	.731	**.745**	.743	.727	.734	**.577**	.558	**.577**	**.577**	.571	**.577**
	gpt2-medium	.782	.766	**.792**	.782	.786	.759	.727	.737	.749	.727	**.757**	.734	.566	.566	.554	.566	.587	**.588**
	gpt2-large	.753	.757	**.805**	.753	.778	.759	.719	.726	.748	.719	**.766**	.732	.564	.532	.549	.564	.543	**.568**
	opt-1.3b	.799	.782	**.826**	.799	.776	.755	.751	.750	**.766**	.751	**.766**	.749	.535	.547	.544	.535	.547	**.574**
	SmolLM-135M	.712	.722	.718	.712	**.753**	.674	**.736**	.730	.726	**.736**	.725	.709	.563	.554	.573	.563	.576	**.595**
	SmolLM-360M	.790	**.805**	.772	.790	.764	.741	.736	**.754**	.752	.736	.735	.733	**.602**	.577	.587	**.602**	.585	.557
	SmolLM-1.7B	**.809**	.792	.807	**.809**	.801	.753	.750	.752	.761	.750	**.766**	.743	.543	.543	.535	.543	**.553**	.545
	pythia-1.4b	.778	.788	**.790**	.778	.774	.732	.734	.770	**.773**	.734	.768	.742	.552	.548	**.562**	.552	.532	.537
	Qwen3-1.7B-Base	.809	.797	.805	.809	**.834**	.763	.759	.761	**.775**	.759	.746	.761	.523	.538	**.576**	.523	.553	.553
$A_{\text{comp}} \uparrow$	gpt2	.768	.823	**.849**	.768	**.849**	.754	.657	.664	.677	.657	**.685**	.656	.663	.661	.658	.663	**.666**	.658
	gpt2-medium	.816	.836	**.875**	.816	.873	.801	.659	.656	**.676**	.659	.654	.658	**.673**	.657	.661	**.673**	.667	.660
	gpt2-large	.810	.857	**.894**	.810	.884	.810	.633	.642	**.657**	.633	.633	.629	**.656**	.644	.647	**.656**	.640	.631
	opt-1.3b	.847	.878	.895	.847	**.898**	.831	.632	.630	.647	.632	**.650**	.627	**.650**	.618	.620	**.650**	.613	.616
	SmolLM-135M	.817	.810	**.829**	.817	.798	.730	.669	.690	**.696**	.669	.679	.663	.666	.668	**.672**	.666	.665	.663
	SmolLM-360M	.799	.808	.827	.799	**.837**	.806	.647	.655	**.658**	.647	.656	.649	.667	**.674**	.669	.667	.669	.662
	SmolLM-1.7B	.837	.852	.883	.837	**.896**	.857	.627	.630	**.647**	.627	.643	.636	.646	.636	**.649**	.646	**.649**	.639
	pythia-1.4b	.834	.873	**.897**	.834	.872	.797	.636	**.647**	.640	.636	**.647**	.628	**.639**	.621	.628	**.639**	.625	.625
	Qwen3-1.7B-Base	.845	.854	**.894**	.845	.893	.842	.632	.617	**.644**	.632	.643	.626	.629	**.645**	.640	.629	**.645**	.626
$V_\ell \uparrow$	gpt2	.834	**.959**	.683	.477	.887	.456	.674	**.831**	.768	.554	.608	.505	**.689**	.586	.547	.523	.572	.538
	gpt2-medium	.885	**.967**	.665	.503	.913	.461	.738	**.855**	.773	.564	.625	.492	**.770**	.629	.576	.523	.582	.503
	gpt2-large	.896	**.970**	.646	.489	.933	.466	.740	**.862**	.782	.572	.643	.503	**.775**	.626	.568	.552	.566	.490
	opt-1.3b	.926	**.977**	.645	.514	.938	.462	.764	**.863**	.791	.583	.625	.502	**.799**	.582	.568	.590	.545	.503
	SmolLM-135M	.856	**.906**	.765	.487	.664	.431	.686	**.760**	.705	.568	.553	.493	.727	**.760**	.715	.545	.553	.523
	SmolLM-360M	.886	**.947**	.789	.493	.754	.455	.727	**.787**	.730	.571	.604	.503	.752	**.776**	.726	.540	.575	.518
	SmolLM-1.7B	.915	**.974**	.616	.528	.949	.440	.783	**.861**	.767	.605	.660	.506	**.793**	.745	.672	.586	.644	.516
	pythia-1.4b	.936	**.970**	.664	.541	.936	.447	.755	**.858**	.774	.594	.620	.507	**.786**	.594	.552	.553	.522	.509
	Qwen3-1.7B-Base	.914	**.976**	.647	.582	.955	.421	.773	**.862**	.759	.601	.666	.496	.758	**.777**	.665	.581	.673	.497
$D_\ell \uparrow$	gpt2	.271	.201	.476	.139	.270	**.729**	.332	.158	**.432**	.146	.132	.351	.656	.669	.725	.343	.635	**.728**
	gpt2-medium	.291	.206	.395	.132	.253	**.771**	.278	.132	**.406**	.173	.120	.365	.641	.556	.662	.390	.654	**.807**
	gpt2-large	.197	.197	.405	.122	.252	**.783**	.266	.114	**.413**	.137	.115	.343	.588	.604	**.728**	.327	.592	.682
	opt-1.3b	.202	.170	.419	.130	.240	**.771**	.213	.101	**.380**	.157	.115	.353	.583	.628	.679	.347	.563	**.694**
	SmolLM-135M	.156	.214	.328	.139	.330	**.492**	.282	.245	**.398**	.166	.217	.311	.574	.592	.639	.393	.713	**.746**
	SmolLM-360M	.161	.168	.296	.165	.357	**.623**	.231	.190	.345	.171	.251	**.392**	.593	.624	.662	.460	.654	**.714**
	SmolLM-1.7B	.160	.165	.415	.170	.243	**.801**	.195	.117	**.371**	.182	.116	.347	.549	.439	.527	.566	.645	**.843**
	pythia-1.4b	.145	.161	.399	.163	.257	**.748**	.175	.108	**.376**	.192	.109	.322	.571	.590	**.696**	.466	.539	.624
	Qwen3-1.7B-Base	.219	.152	.435	.297	.252	**.783**	.188	.106	**.379**	.301	.109	.361	.578	.509	.608	.722	.752	**.923**
bias $\leftarrow$	gpt2	.0276	.0343	.0163	.0458	.0256	**.0100**	.0391	.0501	**.0376**	.0669	.1464	.1441	.0379	.0365	.0299	.0914	.0400	**.0289**
	gpt2-medium	.0297	.0357	.0177	.0439	.0268	**.0098**	.0426	.0558	**.0406**	.0618	.1457	.1503	.0418	.0486	.0358	.0840	.0383	**.0228**
	gpt2-large	.0355	.0366	.0175	.0483	.0262	**.0104**	.0443	.0586	**.0424**	.0701	.1382	.1705	.0454	.0425	**.0307**	.1119	.0491	.0382
	opt-1.3b	.0351	.0374	.0176	.0433	.0275	**.0107**	.0469	.0621	**.0438**	.0613	.1576	.1726	.0489	.0423	**.0334**	.0835	.0512	.0373
	SmolLM-135M	.0412	.0309	.0206	.0431	.0197	**.0140**	.0420	.0418	**.0338**	.0528	.0844	.0870	.0457	.0421	.0364	.0780	.0333	**.0282**
	SmolLM-360M	.0438	.0357	.0227	.0403	.0191	**.0111**	.0472	.0466	**.0381**	.0531	.0731	.0745	.0449	.0424	.0372	.0661	.0371	**.0306**
	SmolLM-1.7B	.0420	.0402	.0188	.0365	.0250	**.0098**	.0498	.0592	**.0448**	.0479	.1376	.1901	.0548	.0727	.0508	.0549	.0395	**.0180**
	pythia-1.4b	.0421	.0393	.0176	.0387	.0289	**.0130**	.0547	.0631	**.0461**	.0529	.1913	.2426	.0456	.0481	**.0326**	.0632	.0530	.0394
	Qwen3-1.7B-Base	.0367	.0379	.0174	.0216	.0238	**.0083**	.0559	.0613	.0447	**.0384**	.1644	.1960	.0511	.0570	.0430	.0307	.0305	**.0111**
pref. $\rightarrow .5$	gpt2	.615	.660	.526	**.478**	.675	.468	.525	.560	.480	.513	.523	**.496**	**.510**	.558	.530	.515	.457	.475
	gpt2-medium	.596	.656	.530	.534	.623	**.472**	.530	.554	.472	.527	.527	**.495**	.508	.573	.530	.508	**.503**	**.503**
	gpt2-large	.629	.654	**.515**	.540	.623	.456	**.501**	.545	.458	.517	.521	.489	.523	.470	**.500**	.525	.475	.475
	opt-1.3b	.634	.692	.447	.551	.663	**.460**	.515	.543	.448	**.505**	.513	.490	.513	.520	.533	**.503**	.518	.480
	SmolLM-135M	.557	.652	.573	**.505**	.605	.509	**.500**	.543	.490	.498	.523	.501	.510	.561	.533	.472	.437	**.500**
	SmolLM-360M	.547	.658	.561	.538	.613	**.530**	.513	.552	.492	.512	.523	**.502**	.510	.505	.533	**.503**	.533	.553
	SmolLM-1.7B	.640	.663	**.507**	.530	.658	.462	.510	.542	.472	**.499**	.527	.498	**.490**	.535	.528	.482	.444	.452
	pythia-1.4b	.627	.662	.513	.557	.636	**.497**	.519	.543	.463	.510	.518	**.496**	.487	.530	.513	.487	**.495**	.457
	Qwen3-1.7B-Base	.588	.681	.524	**.487**	.638	.431	.513	.537	.464	**.504**	.521	.492	.490	.520	**.492**	.477	.480	.429

B.5 Probe-guided steering

Table 18 reports steering results for GPT-2, SmolLM-135M, OPT-1.3B and Qwen3-1.7B-Base at the development-selected dose σ_z and the first input ordering. The original primary comparison is the GPT-2 directly connected-pair likelihood response under unshuffling at $k = \text{all}$ minus its matched random-edit response; OPT and Qwen form the later model extension. Panel (a) places raw likelihood responses beside their paired differences from random edits; panel (b) reports raw generation contrasts. Likelihood responses score the supplied two-step continuations, whereas S^+, S^- and Δe measure sampled generation.

SmolLM-135M's directly connected-pair forward responses under unshuffling at $k = \text{all}$ and $k = 1$ exceed both orthogonal and shuffled-label controls. Their raw responses and differences from

random edits remain positive, with intervals excluding zero, at all three tested doses and under the second input ordering. Neither OPT nor Qwen has a trained forward-target setting with both positive intervals in these robustness checks. Qwen's reverse-target response under unshuffling at $k = 1$ on directly connected pairs is -0.251 ± 0.218, with a difference from random edits of -0.223 ± 0.215 in 10^{-3} nats/token. Its difference from orthogonal edits also has the predicted negative sign with an interval excluding zero; its shuffled-label comparison includes zero.

For generated order, every trained setting's S^+ and S^- intervals include zero, both raw and relative to all three control families. The SmolLM-135M target-error contrast under unshuffling at $k = 1$ is 0.195 ± 0.343 percentage points before control subtraction and 0.195 ± 0.171 relative to random edits. Every trained setting's Δe contrast against orthogonal and shuffled-label edits includes zero.

All trained settings meet the validity-preservation criterion: each of the four real edits has a lower 95% bound on validity change above -2 percentage points. This criterion concerns changes from each setting's own baseline validity; the pretrained baseline validity was zero. These findings concern the specified interventions and do not establish a general absence of causal influence.

Table 18: Probe-guided steering on Blocks World. Training settings use the unshuffling format; Control denotes invalid-target training. (a) Raw forward-target likelihood responses and paired differences from random edits, in 10^{-3} nats/token. (b) Raw generated-order contrasts S^+ and S^- on incomparable pairs and target-edge error contrast Δe on directly connected pairs, in percentage points. Headers give predicted signs. Entries are five-seed means $\pm$ 95% Student-t half-widths without multiplicity adjustment, at dose σ_z and the first input ordering. Bold marks column maxima across training settings within each model, including Control and ties; S^- uses minima. Likelihood uses 256 plans per pair type per seed; generation uses 128 and four draws per condition, with all outputs in the denominator, including invalid and malformed outputs. Plain 0 denotes an exactly zero observed contrast in every seed, not a population-zero bound.

(a) Likelihood responses ΔM_{uv} (10^{-3} nats/token; predicted > 0)

	Directly connected		Incomparable	
Unshuffling	Raw	Real − random	Raw	Real − random
GPT-2				
$k = $ all	$\mathbf{0.039 \pm 0.288}$	$\mathbf{0.046 \pm 0.291}$	0.061 ± 0.026	0.066 ± 0.025
$k = 1$	-0.045 ± 0.455	-0.058 ± 0.497	$\mathbf{0.176 \pm 0.319}$	$\mathbf{0.162 \pm 0.310}$
Control	-0.002 ± 0.006	-0.003 ± 0.006	$-2.82\times10^{-4} \pm 0.003$	$-2.35\times10^{-4} \pm 0.003$
SmolLM-135M				
$k = $ all	$\mathbf{1.564 \pm 0.505}$	$\mathbf{1.752 \pm 0.514}$	$\mathbf{0.329 \pm 0.355}$	$\mathbf{0.355 \pm 0.356}$
$k = 1$	0.973 ± 0.141	1.091 ± 0.129	0.141 ± 0.220	0.259 ± 0.221
Control	0.085 ± 0.034	0.101 ± 0.035	-0.155 ± 0.014	-0.152 ± 0.014
OPT-1.3B				
$k = $ all	$\mathbf{0.001 \pm 0.023}$	$\mathbf{0.001 \pm 0.028}$	-0.003 ± 0.009	-0.003 ± 0.009
$k = 1$	-0.011 ± 0.011	-0.014 ± 0.012	-0.002 ± 0.021	$\mathbf{0.001 \pm 0.019}$
Control	$-4.62\times10^{-5} \pm 4.30\times10^{-4}$	$-1.52\times10^{-5} \pm 4.02\times10^{-4}$	$\mathbf{5.55\times10^{-5} \pm 2.73\times10^{-4}}$	$8.49\times10^{-5} \pm 2.85\times10^{-4}$
Qwen3-1.7B-Base				
$k = $ all	$\mathbf{0.013 \pm 0.042}$	$\mathbf{0.008 \pm 0.045}$	-0.001 ± 0.003	-0.001 ± 0.003
$k = 1$	-0.013 ± 0.229	0.002 ± 0.233	-0.021 ± 0.069	-0.035 ± 0.067
Control	$3.81\times10^{-5} \pm 1.02\times10^{-4}$	$3.72\times10^{-5} \pm 9.52\times10^{-5}$	$\mathbf{-7.73\times10^{-5} \pm 9.48\times10^{-5}}$	$\mathbf{-1.07\times10^{-4} \pm 1.13\times10^{-4}}$

(b) Generation responses (percentage points)

	Order, positive	Order, negative	Target errors
Unshuffling	$S^+ > 0$	$S^- < 0$	$\Delta e > 0$
GPT-2			
$k = $ all	$\mathbf{0.039 \pm 0.108}$	$\mathbf{-0.059 \pm 0.108}$	$\mathbf{0}$
$k = 1$	-0.078 ± 0.217	0 ± 0.171	$\mathbf{0}$
Control	0	0	$\mathbf{0}$
SmolLM-135M			
$k = $ all	$\mathbf{0 \pm 0.086}$	0 ± 0.210	0 ± 0.343
$k = 1$	-0.020 ± 0.054	$\mathbf{-0.020 \pm 0.133}$	$\mathbf{0.195 \pm 0.343}$
Control	$\mathbf{0}$	0	0.078 ± 0.133
OPT-1.3B			
$k = $ all	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$
$k = 1$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$
Control	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$
Qwen3-1.7B-Base			
$k = $ all	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$
$k = 1$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$
Control	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$