TODAY'S TARGETS, TOMORROW'S CONTEXT: MEASURING TOKEN-LEVEL FUTURE INFLUENCE IN SFT

Anonymous authors
Paper under double-blind review

ABSTRACT

Reasoning supervised fine-tuning (SFT) pipelines typically filter complete trajectories by final-answer correctness, then directly supervise every response token in the retained trajectories. Yet final-answer correctness does not imply that every token should receive direct supervision. In autoregressive generation, a token is both a prediction target and, once generated, context for subsequent predictions; supervising it therefore also affects subsequent generation. Even in a trajectory reaching the correct answer, some tokens, as context, may make the fixed demonstrated continuation harder for the current model to predict. We introduce *Future Influence*, which measures the direction of this contextual effect and uses it to determine whether to retain direct supervision. Keeping the demonstrated sequence unchanged, we attenuate the information a token transmits to later positions: an increase in continuation loss indicates that the token supports prediction, whereas a decrease indicates that it makes prediction harder. Using differentiable gates and a single backward pass, we obtain a first-order approximation to these intervention-induced loss changes, simultaneously estimating the directions of all response tokens. During training, we mask only the direct prediction losses of negatively directed tokens, retaining them as context. Experiments show that negatively directed tokens are common even in reasoning trajectories with correct final answers. Compared with standard SFT, supervision selected by Future Influence improves performance across model scales, training datasets, and mathematical reasoning benchmarks.

1 INTRODUCTION

Many reasoning SFT datasets follow a similar construction process: complete reasoning trajectories are sampled from a model, filtered by final-answer correctness, and retained for training (Zelikman et al., 2022; Yuan et al., 2023; DeepSeek-AI, 2025). The broader practice of using model-generated data to improve models also appears in self-training research (Gulcehre et al., 2023; Huang et al., 2022) and recent reasoning corpora (Yu et al., 2024; Ye et al., 2025; Muennighoff et al., 2025; Guha et al., 2025). In this pipeline, verification is performed at the trajectory level, whereas the SFT prediction loss is applied at the token level: once a trajectory is retained, its response tokens are typically included in the training objective. Yet final-answer correctness does not establish that every local token transition should be directly reinforced. A token may lie on a correct trajectory, while the predictive state induced by its contextual role makes it harder for the current model to continue the demonstrated trajectory.

In autoregressive modeling, each response token has two roles: under teacher-forced SFT, it is the prediction target at its own position; after generation, it becomes context for subsequent predictions. Prior work on exposure bias has shown that local deviations during generation can propagate through subsequent contexts and accumulate (Bengio et al., 2015; Ranzato et al., 2016; Wiseman & Rush, 2016; Ross et al., 2011; Arora et al., 2022). Thus, a reasoning trajectory is both a sequence of prediction targets and a sequence of predictive states induced by its prefixes. Each local token transition forms the context for the next prediction, and the resulting state affects whether the model can continue the demonstrated trajectory. We therefore ask which local transitions lead the model into states that support the demonstrated continuation. Today's target is tomorrow's context.

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.

Prior work adjusts token-level supervision in SFT using signals such as target-distribution alignment, token difficulty, training dynamics, semantic information, and gradient alignment for target adaptation and capability retention (Lin et al., 2025; Fu et al., 2025; Qin et al., 2025; Pang et al., 2025; **?**). We instead examine the contextual state induced by each token along a fixed demonstrated trajectory. Keeping the model parameters, input sequence, token positions, and demonstrated targets fixed, we attenuate only the information transmitted from a token through the value pathway to later positions, and compare the prediction loss on the fixed continuation before and after the intervention. If the loss increases, the original information flow supports the continuation; if the loss decreases, weakening it makes the continuation easier to predict. We call this signed contextual effect under the specified intervention Future Influence, and use it as an operational signal of whether the current predictive state supports the demonstrated continuation.

From this state perspective, we retain the direct prediction loss for tokens with nonnegative Future Influence and mask it for tokens with negative influence. The full demonstrated sequence remains unchanged, so masked tokens continue to serve as context for later predictions. Directly measuring each token's direction requires separately weakening its information flow and recomputing the loss on the fixed continuation. The number of additional forward passes grows linearly with response length, making this computation difficult to apply at scale to typically long CoT responses. We therefore introduce a differentiable gate for each response token to control the information available to later positions, and use the first-order derivative of the response loss with respect to each gate to approximate the corresponding intervention-induced loss change. A single forward pass and a single backward pass can thus estimate the directions of all response tokens simultaneously; experiments verify that the gate-gradient directions agree with those obtained by direct information intervention.

Overall, for complete demonstrated trajectories that pass answer-level verification, we decide whether to retain each token's direct supervision according to whether its contextual transition supports the demonstrated continuation. Tokens with negative Future Influence are common even in reasoning trajectories with correct final answers. Compared with standard SFT, masking their direct supervision improves performance across model scales and training datasets, while retaining tokens with the opposite direction gives the weakest overall results. Based on these results, our contributions are threefold:

1. We introduce Future Influence, using the direction of a token's contextual effect on prediction of the fixed demonstrated continuation as a criterion for selecting direct supervision.
2. We develop a gate-gradient approximation that simultaneously estimates the directions of all response tokens in a sequence with a single forward pass and a single backward pass.
3. We demonstrate across model scales and training datasets that masking the direct supervision of negatively directed tokens improves performance over standard SFT.

2 METHOD

Given a complete demonstrated trajectory, we measure the direction of each response token's contextual effect on the predictive state used to follow the fixed demonstrated continuation. We then use this direction to decide whether to retain the token's direct SFT supervision. Throughout the procedure, the demonstrated sequence remains unchanged: tokens whose direct supervision is masked stay in the sequence and continue to serve as context for subsequent predictions. Figure 1 gives an overview.

2.1 PROBLEM SETUP: TARGET AND CONTEXT ROLES

Given a prompt x and a complete demonstrated response $y_{1:T}$ of length T, let p_θ denote an autoregressive model with parameters θ, and let $y_{<i}$ denote the response-token prefix before position i. The model predicts token y_i from x and $y_{<i}$. Under teacher-forced SFT, y_i is the prediction target at position i; for later targets, it becomes part of the prefix that determines the next predictive state. We therefore measure how y_i, as context, affects the model's ability to follow the fixed demonstrated continuation.

Let $\mathcal{C}$ be the index set of response-content tokens whose directions are estimated, and let $\mathcal{R}$ be the index set of response targets included in the response loss, with $\mathcal{C} \subseteq \mathcal{R}$. These indices refer to

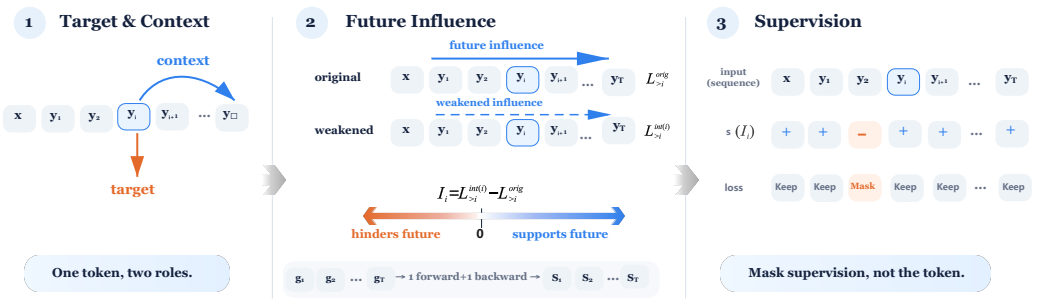


Figure 1: Overview. **Left:** a response token plays two roles: the direct prediction target at its own position and context for the demonstrated continuation that follows it. **Middle:** we weaken the token's information through the specified value pathway while keeping the sequence, positions, and demonstrated targets unchanged, and measure the resulting change in loss on the future targets; a first-order estimate from a single backward pass gives this direction for all tokens at once. **Right:** no token is removed from the input. Only direct supervision is withheld from tokens whose contextual direction is negative.

positions in the response sequence, rather than positions in the concatenated input. The distinction is that $\mathcal{R}$ may include the response end marker, whereas the end marker is not a response-content token in $\mathcal{C}$ and therefore does not receive a separately estimated direction. For any $i \in \mathcal{C}$, define the future target set of y_i as

$$\mathcal{F}(i) = \{j \in \mathcal{R} : j > i\}. \tag{1}$$

That is, $\mathcal{F}(i)$ contains all demonstrated targets after y_i that contribute to the response loss. If the response end marker is included in $\mathcal{R}$, it is also a future target of the final response-content token. In the remainder of this section, we compare prediction loss before and after information-flow intervention on this fixed future-target set.

2.2 Future Influence: An Information-Flow Counterfactual on the Fixed Demonstrated Continuation

To measure the effect of y_i as context for subsequent predictions, we compare the model's prediction loss on the same demonstrated future targets before and after weakening its value-path information. Under the original information flow, the loss on its future-target set $\mathcal{F}(i)$ is

$$L_{>i}^{\text{orig}} = -\sum_{j \in \mathcal{F}(i)} \log p_\theta(y_j \mid x, y_{<j}). \tag{2}$$

We then intervene on the contextual information that y_i transmits to later positions through the specified value pathway, while keeping the model parameters, input token sequence, token positions, and all demonstrated targets unchanged. The intervention weakens this value-path information flow inside the model; it does not delete or replace y_i in the input. Let $p_\theta^{\text{int}(i)}$ denote the predictive distribution under this intervention. The corresponding future loss is

$$L_{>i}^{\text{int}(i)} = -\sum_{j \in \mathcal{F}(i)} \log p_\theta^{\text{int}(i)}(y_j \mid x, y_{<j}). \tag{3}$$

Both evaluations use exactly the same demonstrated prefixes and future targets; they differ only in how the value-path information from y_i is transmitted while the model processes those prefixes. The loss change therefore measures whether the original contextual state supports the fixed demonstrated continuation.

We define the **Future Influence** of token y_i as the signed change in future loss:

$$I_i = L_{>i}^{\text{int}(i)} - L_{>i}^{\text{orig}}. \tag{4}$$

If $I_i > 0$, weakening the value-path information from y_i increases future loss, indicating that the corresponding contextual state supports prediction of the fixed demonstrated continuation. If $I_i < 0$, weakening that information decreases future loss, indicating that the corresponding contextual state makes the continuation harder for the current model to predict. If $I_i = 0$, the intervention leaves the total prediction loss on the future targets unchanged.

We focus on the *direction* of this contextual effect, given by the sign of I_i. The next section describes how gate gradients efficiently estimate these directions for all response-content tokens.

2.3 Efficient Estimation with Gate Gradients

Directly computing Future Influence requires a separate value-path intervention and a new evaluation of the demonstrated future loss for each response-content token. For long reasoning trajectories, this entails an additional forward pass for every token. Because an intervention on y_i can affect only later response targets, its future-loss change can be written as a change in the full response loss. We then estimate the directions of these changes jointly using gate gradients.

A unified response-loss objective. Define the prediction loss over the full response as

$$L_{\text{resp}} = -\sum_{j \in \mathcal{R}} \log p_\theta(y_j \mid x, y_{<j}). \tag{5}$$

In autoregressive prediction, the probability of y_i is computed from x and $y_{<i}$, before the model reads y_i. Intervening on the value-path information from y_i therefore cannot change the prediction of y_i or any earlier target; it can affect only subsequent targets. The terms in the full response loss for positions at or before i remain unchanged under the intervention, giving

$$I_i = L_{\text{resp}}^{\text{int}(i)} - L_{\text{resp}}^{\text{orig}}. \tag{6}$$

This equality lets us measure the contextual effects of different tokens with the same full-response objective, without constructing a separate future-loss objective for each token.

Direction estimation with differentiable gates. We introduce a differentiable scalar gate g_i for each response-content token to control the contextual information it passes to later positions. A value of $g_i = 1$ corresponds to the original information flow, while decreasing g_i weakens it. The gates form a vector $g = (g_i)_{i \in \mathcal{C}}$, and we denote the corresponding full response loss by $L_{\text{resp}}(g)$. The specific gate placement within the model is described in Appendix B.3.

Starting from $g = \mathbf{1}$, we decrease g_i by a small positive amount ϵ while keeping all other gates at one. A first-order expansion gives

$$L_{\text{resp}}(g_i = 1 - \epsilon) - L_{\text{resp}}(g_i = 1) \approx -\epsilon \left. \frac{\partial L_{\text{resp}}(g)}{\partial g_i} \right|_{g=\mathbf{1}}. \tag{7}$$

Because $\epsilon > 0$, the direction of this local loss change is determined by the negative gate gradient. We therefore define the signed direction estimate

$$s_i = -\left. \frac{\partial L_{\text{resp}}(g)}{\partial g_i} \right|_{g=\mathbf{1}}. \tag{8}$$

When $s_i > 0$, locally weakening the value-path information from y_i increases future loss; when $s_i < 0$, it decreases future loss. Since g_i does not affect targets at or before position i, its gradient receives loss contributions only from subsequent response targets and estimates the direction of the corresponding contextual state transition.

With all gates initialized to one, a single forward pass and a single backward pass yield s_i for all response-content tokens simultaneously, without a separate intervention forward pass for each token. These local gradients provide a first-order estimate of the direction of each token-induced state transition. The next section uses their signs to construct the direct-supervision mask, and experiments evaluate their agreement with directions obtained by direct intervention.

2.4 SIGN-BASED MASKED SFT

Under teacher forcing, retaining the direct loss of y_i reinforces the model's prediction of that token. Once generated, however, y_i becomes part of the context for later predictions and induces the next predictive state. We therefore retain direct supervision when the corresponding contextual transition supports the demonstrated continuation, and remove only the direct target loss when it makes the continuation harder to predict.

We convert these directions into a binary supervision mask,

$$m_i = \mathbb{1}\{s_i \geq 0\}, \qquad i \in \mathcal{C}. \tag{9}$$

For $m_i = 1$, the direct prediction loss of y_i is retained; for $m_i = 0$, only that target loss is removed. In both cases, the complete demonstrated sequence remains in the teacher-forced input, so a token with masked direct supervision continues to serve as context for later predictions. The method therefore selects which local transitions receive direct reinforcement while preserving the full demonstrated context.

Writing $w_j = m_j$ for $j \in \mathcal{C}$ and $w_j = 1$ for other response targets, the masked SFT loss is

$$L_{\mathrm{SFT}}^{\mathrm{mask}}(\theta) = -\sum_{j \in \mathcal{R}} w_j \log p_\theta(y_j \mid x, y_{<j}). \tag{10}$$

The end marker remains a supervised target but is not assigned a content-token direction. Example-wise and batch-wise loss normalization is described in Appendix B.2. The zero boundary of the signed direction determines whether each content target receives direct supervision, without top-k ranking, a preset retention ratio, or continuous loss weights.

Algorithm 1 summarizes the offline scoring and masked-SFT stages. Using the same frozen base checkpoint that initializes SFT, we estimate the direction of each response-content token and construct supervision labels solely from its sign. The model is then trained on the complete demonstrated sequences, retaining direct supervision only for selected targets. The supervision masks are computed once before training and remain fixed throughout SFT.

Algorithm 1: Future-Influence Masked SFT

Input: Dataset $\mathcal{D}$, frozen base model θ_0.
Output: Fine-tuned model θ.

Initialize trainable model $\theta \leftarrow \theta_0$;
foreach $(x, y) \in \mathcal{D}$ **do**
  Identify content indices $\mathcal{C}$ and response targets $\mathcal{R}$;
  Initialize all token gates $g \leftarrow \mathbf{1}$;
  Forward once with frozen θ_0: $L \leftarrow L_{\mathrm{resp}}(g; \theta_0)$;
  Backward once for token directions: $s \leftarrow -\nabla_g L$;
  Build the supervision mask: $m_i \leftarrow \mathbb{1}\{s_i \geq 0\}, \quad i \in \mathcal{C}$;
  Ignore target labels where $m_i = 0$; retain EOS labels;
  Store labels; keep input tokens intact;
Fine-tune θ with fixed stored labels, minimizing $L_{\mathrm{SFT}}^{\mathrm{mask}}$;
return θ;

3 EXPERIMENTS

3.1 SETUP

Data. We use two mathematical reasoning CoT datasets, OpenR1-Math-220k (Open R1, 2025) and OpenMathReasoning (Moshkov et al., 2025). From each dataset, we randomly sample 100,000 trajectories with correct final answers and sequence lengths below 16k tokens for training. The two corpora are used independently and never mixed; all compared methods use the same training samples within each corpus. Dataset descriptions and training-subset construction are provided in Appendix D.

Models. We use Qwen3-4B-Base, Qwen3-8B-Base, and Qwen3-14B-Base (Yang et al., 2025a) to evaluate across model scales, and Gemma3-12B (Gemma Team et al., 2025) to evaluate across model families. Each model undergoes full-parameter SFT separately on the two corpora. Detailed training settings are provided in Appendix B.

Baselines. All methods are trained on the same data with the same hyperparameters and evaluated under the same protocol. (1) **SFT** supervises all response targets. (2) **AlphaToken (?)** selects tokens

Table 1: pass@32 on six competition-level benchmarks for models trained on OpenR1-Math-220k and OpenMathReasoning. General is the unweighted mean of IFEval (strict), TruthfulQA MC1 and MC2, and ARC-Challenge accuracy and length-normalized accuracy, and is reported only as supplementary OOD evidence. HMMT26 and HMMT25 denote HMMT February 2026 and HMMT November 2025.

	OpenR1-Math-220k							OpenMathReasoning						
Method	AIME24	AIME25	AIME26	AMC23	HMMT26	HMMT25	Gen.	AIME24	AIME25	AIME26	AMC23	HMMT26	HMMT25	Gen.
Qwen3-4B-Base														
SFT	50.00	40.00	40.00	95.00	27.27	26.67	37.98	46.67	43.33	40.00	90.00	24.24	23.33	35.77
DFT	50.00	40.00	36.67	95.00	27.27	30.00	38.15	50.00	46.67	43.33	90.00	24.24	23.33	36.06
AlphaToken	53.33	43.33	40.00	97.50	24.24	30.00	37.92	50.00	46.67	40.00	95.00	27.27	26.67	35.83
Cleaning	46.67	46.67	40.00	92.50	30.30	23.33	37.69	43.33	40.00	40.00	87.50	21.21	20.00	35.53
ssToken	53.33	40.00	43.33	95.00	30.30	26.67	38.17	46.67	43.33	46.67	90.00	24.24	23.33	36.05
Random	46.67	40.00	40.00	90.00	27.27	26.67	38.37	43.33	36.67	46.67	87.50	21.21	26.67	36.57
Low-NLL	46.67	40.00	40.00	92.50	27.27	30.00	38.49	43.33	43.33	43.33	92.50	27.27	26.67	35.97
Reverse	46.67	36.67	36.67	87.50	21.21	23.33	38.06	33.33	33.33	26.67	85.00	18.18	20.00	36.34
Ours	**56.67**	**50.00**	**56.67**	**100.00**	**33.33**	**33.33**	37.40	**56.67**	**50.00**	**50.00**	**97.50**	**30.30**	**36.67**	37.06
Qwen3-8B-Base														
SFT	63.33	43.33	56.67	95.00	30.30	43.33	34.86	63.33	43.33	60.00	95.00	30.30	36.67	33.67
DFT	66.67	46.67	56.67	95.00	27.27	43.33	35.15	66.67	43.33	60.00	95.00	30.30	36.67	33.90
AlphaToken	66.67	43.33	60.00	92.50	33.33	40.00	34.72	66.67	43.33	60.00	92.50	33.33	33.33	33.82
Cleaning	60.00	46.67	56.67	95.00	30.30	46.67	34.45	60.00	46.67	56.67	95.00	30.30	40.00	33.48
ssToken	63.33	40.00	53.33	97.50	27.27	43.33	35.01	63.33	40.00	53.33	97.50	27.27	36.67	33.98
Random	63.33	43.33	56.67	97.50	30.30	36.67	34.13	60.00	43.33	60.00	95.00	30.30	36.67	33.96
Low-NLL	63.33	43.33	53.33	95.00	33.33	43.33	34.63	66.67	46.67	56.67	95.00	30.30	36.67	34.77
Reverse	56.67	43.33	46.67	95.00	27.27	33.33	34.29	53.33	36.67	43.33	95.00	21.21	30.00	34.17
Ours	**70.00**	**50.00**	**63.33**	**100.00**	**36.36**	**50.00**	33.70	**70.00**	**50.00**	**63.33**	**100.00**	**36.36**	**46.67**	34.30
Qwen3-14B-Base														
SFT	70.00	56.67	70.00	95.00	36.36	60.00	37.30	73.33	60.00	73.33	90.00	33.33	53.33	36.63
DFT	73.33	60.00	70.00	97.50	36.36	60.00	37.59	76.67	60.00	73.33	92.50	30.30	53.33	36.92
AlphaToken	73.33	60.00	66.67	97.50	36.36	56.67	38.73	76.67	63.33	70.00	95.00	33.33	56.67	36.88
Cleaning	66.67	60.00	70.00	92.50	33.33	60.00	38.33	70.00	60.00	70.00	92.50	30.30	50.00	36.42
ssToken	70.00	63.33	66.67	95.00	33.33	56.67	39.04	76.67	63.33	70.00	95.00	33.33	60.00	37.13
Random	70.00	56.67	70.00	97.50	36.36	60.00	38.52	73.33	53.33	70.00	92.50	33.33	53.33	36.48
Low-NLL	73.33	60.00	70.00	97.50	36.36	60.00	38.78	73.33	53.33	73.33	95.00	33.33	60.00	37.03
Reverse	70.00	60.00	63.33	95.00	33.33	50.00	38.37	66.67	50.00	70.00	90.00	33.33	40.00	36.23
Ours	**83.33**	**66.67**	**73.33**	**100.00**	**39.39**	**63.33**	38.64	**80.00**	**70.00**	**76.67**	**97.50**	**39.39**	**63.33**	37.09
Gemma3-12B														
SFT	30.00	30.00	26.67	87.50	24.24	20.00	34.26	40.00	30.00	26.67	87.50	21.21	20.00	32.80
DFT	30.00	33.33	26.67	87.50	24.24	23.33	34.37	43.33	26.67	30.00	87.50	24.24	23.33	32.97
AlphaToken	33.33	33.33	26.67	87.50	24.24	20.00	34.09	43.33	33.33	26.67	87.50	21.21	23.33	32.71
Cleaning	26.67	30.00	30.00	92.50	27.27	20.00	33.88	36.67	30.00	30.00	92.50	24.24	20.00	32.47
ssToken	30.00	23.33	20.00	85.00	18.18	16.67	34.33	40.00	26.67	23.33	82.50	18.18	16.67	32.89
Random	23.33	26.67	23.33	87.50	21.21	16.67	33.33	33.33	33.33	30.00	90.00	21.21	23.33	32.58
Low-NLL	36.67	33.33	30.00	92.50	24.24	23.33	33.70	43.33	33.33	33.33	92.50	24.24	23.33	32.79
Reverse	20.00	16.67	16.67	80.00	18.18	13.33	32.85	33.33	26.67	33.33	85.00	15.15	13.33	31.94
Ours	**43.33**	**36.67**	**36.67**	**95.00**	**30.30**	**26.67**	33.70	**46.67**	**36.67**	**36.67**	**95.00**	**30.30**	**33.33**	32.58

using path-aware gradient alignment for target adaptation and capability retention. (3) **Token Cleaning** (Pang et al., 2025) performs two-stage fine-grained selection of SFT data. (4) **ssToken** (Qin et al., 2025) selects tokens from training dynamics together with token semantics. (5) **DFT** (**?**) reweights token losses using the model's detached target-token probabilities. (6) **Random masking** masks the same number of tokens per sequence at random, fixing the token budget. (7) **Low-NLL masking** masks the same number of lowest-NLL tokens, using token NLL as the selection criterion. (8) **Reverse masking** masks nonnegative content tokens, including zeros, and retains negative ones without additionally matching token counts. (9) **Ours** masks negatively directed tokens, using only the sign of the direction, with no ranking, retention ratio, or tunable threshold. Shared training settings and comparison methods are described in Appendices B and C.

Evaluation. We evaluate on six competition-level mathematical benchmarks: AIME 2024, AIME 2025, AIME 2026, AMC 2023, HMMT February 2026, and HMMT November 2025; see MathArena (Balunović et al., 2025) for the AIME and HMMT evaluation sets. We sample 32 responses

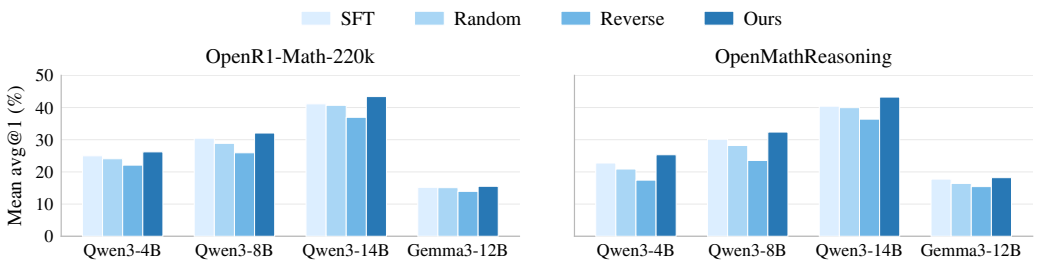


Figure 2: Mean avg@1 over the six mathematical benchmarks for SFT, random masking, reverse masking, and our method. Each bar is the unweighted mean of the six per-benchmark avg@1 scores for a model–corpus–method setting. The panels correspond to the two training corpora; per-benchmark results for all methods are in Appendix H.

per problem and use pass@32 as the primary metric, defined as the fraction of problems with at least one correct response. We also report average single-sample accuracy, avg@1, for each benchmark, with its cross-benchmark mean shown in the main text and per-benchmark results in Appendix H. Additionally, we report IFEval (strict) (Zhou et al., 2023), TruthfulQA MC1 and MC2 (Lin et al., 2021), and ARC-Challenge accuracy and length-normalized accuracy (Clark et al., 2018) to assess instruction following, truthfulness, and science question answering, respectively. Gen. in the main table is the unweighted mean of these five metrics. The full evaluation protocol is provided in Appendix E.

3.2 Main Results

Mathematical reasoning performance. Table 1 reports pass@32 on six competition-level mathematical benchmarks for four models fine-tuned separately on two training corpora. Our method achieves the highest score in all 48 model–corpus–benchmark combinations, outperforming standard SFT, four published token-level supervision methods, and three internal masking controls. The comparison spans full-target supervision, continuous loss reweighting, path-aware gradient alignment, training-dynamics-based selection, and budget-controlled masking. The result holds across all Qwen3 scales, Gemma3-12B, and both training corpora. This consistency shows that the advantage is not tied to a particular model scale, model family, corpus, or baseline mechanism.

Internal masking controls. Random masking and low-NLL masking match our supervision budget separately for each example but choose masked targets randomly or by lowest NLL. Our method outperforms both in all 48 combinations. Because the controls remove the same number of targets from the same examples, these comparisons isolate the selection criterion from the amount of supervision. Random masking tests whether simply reducing direct supervision is sufficient, while low-NLL masking tests whether the gain can be explained by token predictability. Neither control reproduces the result, indicating that the contextual-direction criterion provides information beyond these simpler selection rules.

Direction reversal and single-sample performance. Reverse masking uses the same direction estimate but retains negatively directed targets and masks nonnegative ones. Its mean pass@32 is lowest in all eight model–corpus settings and lowest or tied in 39 of 48 per-benchmark results. Because it reverses the retention rule while using the same estimated directions, this comparison directly tests whether the sign identifies which side should receive supervision. Figure 2 provides a complementary single-sample view: our method has the highest mean avg@1 among SFT, random masking, and reverse masking in all eight settings. Thus, the advantage is not explained by masking or repeated sampling alone; it depends on retaining the direction whose contextual transition supports the demonstrated continuation, and appears in both pass@32 and average single-sample accuracy.

Non-mathematical benchmarks. General capability remains broadly comparable to SFT across IFEval, TruthfulQA, and ARC-Challenge, with detailed per-capability results reported in Appendix G.

Table 2: Sign agreement between the gate-gradient estimate s_i and three single-token interventions, pooled over response-content tokens from 1,000 examples per dataset.

Model	Dataset	g_i=0.99	g_i=0	Removal
Qwen3-14B-Base	OpenR1-Math-220k	0.9976	0.9737	0.9425
Qwen3-14B-Base	OpenMathReasoning	0.9983	0.9815	0.9541
Gemma3-12B	OpenR1-Math-220k	0.9932	0.9728	0.9484
Gemma3-12B	OpenMathReasoning	0.9968	0.9667	0.9525

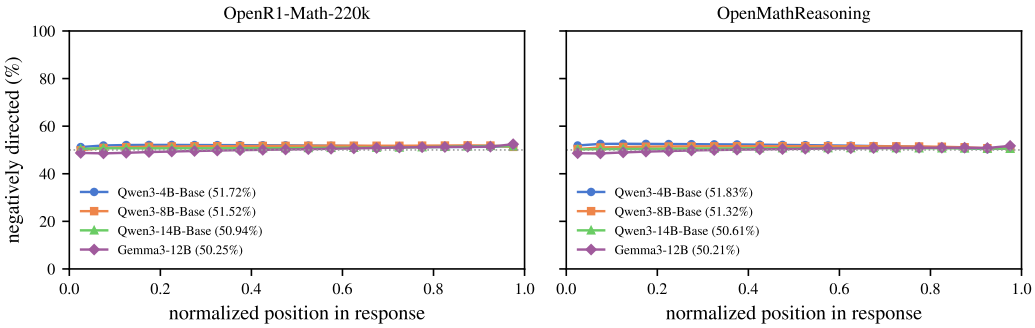


Figure 3: Fraction of negatively directed tokens by normalized position within the response, in 20 equal-width bins, over 100,000 scored samples per model and dataset. Percentages in the legend are the overall fraction for that file. The dotted line marks 50%.

3.3 Validating the Direction Estimate

We compare the direction of the gate-gradient estimate s_i with that obtained by individual token interventions. For Qwen3-14B-Base and Gemma3-12B, we randomly sample 1,000 examples from each dataset and pool all response-content tokens to compute sign agreement. We consider three interventions: small gate attenuation ($g_i = 0.99$), setting the gate to zero ($g_i = 0$), and physical token removal. The first two preserve the demonstrated sequence; the third also changes the sequence and subsequent positions. Table 2 shows a consistent pattern across all four settings: sign agreement is 99.32%–99.83% under small gate attenuation, 96.67%–98.15% when the gate is set to zero, and 94.25%–95.41% under token removal. Agreement remains high even when the input sequence changes, supporting the use of gate gradients to efficiently estimate contextual directions. The full protocol is given in Appendix F.

3.4 Mask Statistics

We analyze supervision masks for three Qwen3 scales and Gemma3-12B on both datasets, covering eight model–corpus settings with 100,000 examples each and approximately 4.669 billion scored response-content tokens in total. Across settings, 50.21%–51.83% of content tokens have negative directions, so the criterion masks direct supervision for about half of the content tokens rather than a small set of outliers. Figure 3 divides each response into 20 equal-width normalized-position bins; the maximum within-setting range is 3.86 percentage points. At this resolution, negatively directed tokens are not markedly concentrated in a particular part of the response. Further statistics are provided in Appendix I.

4 Related Work

Trajectory-level verification and filtering. Reasoning SFT data are commonly constructed by sampling complete trajectories and retaining those whose final answers are correct. Training verifiers from outcome labels offers a related form of answer-level evaluation (Cobbe et al., 2021). STaR (Zelikman et al., 2022) follows this trajectory-level filtering paradigm by bootstrapping rea-

soning traces that reach correct answers. ReST$^{\mathrm{EM}}$ (Gulcehre et al., 2023) similarly generates offline samples from a model and uses them to improve the model. Once a trajectory is retained, its response tokens are typically all used as SFT targets. More broadly, AlpaGasus (Chen et al., 2024) filters instruction data using a strong model, DEITA (Liu et al., 2024) selects instruction examples using data-quality measurements, and Superfiltering (Li et al., 2024) uses a weaker model to filter instruction data for training a stronger model.

Process supervision. Process supervision extends evaluation beyond the final answer to intermediate reasoning steps. Uesato et al. (2022) compare process- and outcome-based feedback for mathematical reasoning. Let's Verify Step by Step (Lightman et al., 2023) uses human judgments of step correctness. Math-Shepherd (Wang et al., 2024) constructs step-level supervision without human annotations. OmegaPRM (Luo et al., 2024) develops automated process supervision for mathematical reasoning. Recent work also examines the design and evaluation of process reward models (Zhang et al., 2025). In contrast, we study how a response token, when used as context, affects the demonstrated continuation that follows it.

Token-level supervision. Recent work adjusts language-model training signals at the token level. Rho-1 (Lin et al., 2025) selects prediction targets according to their alignment with a target distribution; T-SHIRT (Fu et al., 2025), ssToken (Qin et al., 2025), and Token Cleaning (Pang et al., 2025) select tokens using hierarchical data selection, training dynamics and semantics, or fine-grained SFT criteria. AlphaToken (**?**) uses path-aware gradient alignment along direct-update and downstream-causal paths, while DFT (**?**) retains all response targets and continuously reweights their losses using detached target-token probabilities. SePO (Yang et al., 2025b) selects tokens for preference optimization, and unlikelihood training (Welleck et al., 2019) modifies token-level signals by penalizing specified undesirable generations. At coarser granularities, LESS (Xia et al., 2024), DSIR (Xie et al., 2023b), and DoReMi (Xie et al., 2023a) select examples or data mixtures, while Influence Functions (Koh & Liang, 2020), TracIn (Pruthi et al., 2020), Data Shapley (Ghorbani & Zou, 2019), and Integrated Gradients (Sundararajan et al., 2017) assess the effects of training points or input features. We instead start from the predictive states that the model visits along a demonstrated trajectory: once generated, a response token becomes context for subsequent predictions, and the information it transmits may make it easier or harder for the current model to continue the trajectory. We fix the model and demonstrated continuation, weaken the token's information through the value pathway, compare continuation loss before and after the intervention, and use the resulting signed contextual direction to decide whether to retain the token's direct supervision.

5 Conclusion

Answer-level verification confirms that a reasoning trajectory reaches the correct answer, but not whether every token in it is suitable for direct supervision. We introduce Future Influence to measure each token's contextual direction on prediction of the fixed demonstrated continuation, estimating all response-content directions with one forward and one backward pass. We mask direct supervision only for negatively directed tokens while retaining the complete demonstrated context. Across model scales, families, and training corpora, this improves mathematical reasoning pass@32, whereas reversing the direction yields the worst overall results. Budget-matched controls that remove the same number of targets at random or by lowest NLL do not reproduce the gain, so the criterion is not a proxy for the amount of supervision or for token predictability. These findings show that contextual direction provides a useful basis for selecting direct supervision in answer-correct trajectories.

Limitations

Future Influence measures a token's contextual effect on the fixed demonstrated continuation under the scoring model, rather than on every continuation the model might generate. It requires one offline forward-backward scoring pass per sequence before SFT. Our experiments focus on mathematical reasoning trajectories and autoregressive language models; broader domains, data formats, and architectures remain to be studied.

REFERENCES

Kushal Arora, Layla El Asri, Hareesh Bahuleyan, and Jackie Chi Kit Cheung. Why exposure bias matters: An imitation learning perspective of error accumulation in language generation, 2022. URL https://arxiv.org/abs/2204.01171.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions, 2025. URL https://arxiv.org/abs/2505.23281.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks, 2015. URL https://arxiv.org/abs/1506.03099.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. Alpagasus: Training a better alpaca with fewer data, 2024. URL https://arxiv.org/abs/2307.08701.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Yanjun Fu, Faisal Hamman, and Sanghamitra Dutta. T-shirt: Token-selective hierarchical data selection for instruction tuning, 2025. URL https://arxiv.org/abs/2506.01317.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein

Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning, 2019. URL https://arxiv.org/abs/1904.02868.

Etash Guha, Ryan Marten, Sedrick Keh, et al. Openthoughts: Data recipes for reasoning models, 2025. URL https://arxiv.org/abs/2506.04178.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. Reinforced self-training (ReST) for language modeling, 2023. URL https://arxiv.org/abs/2308.08998.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve, 2022. URL https://arxiv.org/abs/2210.11610.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions, 2020. URL https://arxiv.org/abs/1703.04730.

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning, 2024. URL https://arxiv.org/abs/2402.00530.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods, 2021. URL https://arxiv.org/abs/2109.07958.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. Rho-1: Not all tokens are what you need, 2025. URL https://arxiv.org/abs/2404.07965.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning, 2024. URL https://arxiv.org/abs/2312.15685.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, and Abhinav Rastogi. Improve mathematical reasoning in language models by automated process supervision, 2024. URL https://arxiv.org/abs/2406.06592.

Ivan Moshkov, Darragh Hanley, Ivan Sorokin, Shubham Toshniwal, Christof Henkel, Benedikt Schifferer, Wei Du, and Igor Gitman. Aimo-2 winning solution: Building state-of-the-art mathematical reasoning models with openmathreasoning dataset, 2025. URL https://arxiv.org/abs/2504.16891.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025. URL https://arxiv.org/abs/2501.19393.

Open R1. Openr1-math-220k. https://huggingface.co/datasets/open-r1/OpenR1-Math-220k, 2025. Hugging Face dataset.

Jinlong Pang, Na Di, Zhaowei Zhu, Jiaheng Wei, Hao Cheng, Chen Qian, and Yang Liu. Token cleaning: Fine-grained data selection for LLM supervised fine-tuning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 47837–47858. PMLR, 2025. URL https://proceedings.mlr.press/v267/pang25a.html.

Garima Pruthi, Frederick Liu, Mukund Sundararajan, and Satyen Kale. Estimating training data influence by tracing gradient descent, 2020. URL https://arxiv.org/abs/2002.08484.

Xiaohan Qin, Xiaoxing Wang, Ning Liao, Cancheng Zhang, Xiangdong Zhang, Mingquan Feng, Jingzhi Wang, and Junchi Yan. sstoken: Self-modulated and semantic-aware token selection for LLM fine-tuning, 2025. URL https://arxiv.org/abs/2510.18250.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks, 2016. URL https://arxiv.org/abs/1511.06732.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning, 2011. URL https://arxiv.org/abs/1011.0686.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks, 2017. URL https://arxiv.org/abs/1703.01365.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback, 2022. URL https://arxiv.org/abs/2211.14275.

Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations, 2024. URL https://arxiv.org/abs/2312.08935.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training, 2019. URL https://arxiv.org/abs/1908.04319.

Sam Wiseman and Alexander M. Rush. Sequence-to-sequence learning as beam-search optimization, 2016. URL https://arxiv.org/abs/1606.02960.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning, 2024. URL https://arxiv.org/abs/2402.04333.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining, 2023a. URL https://arxiv.org/abs/2305.10429.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling, 2023b. URL https://arxiv.org/abs/2302.03169.

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

Kailai Yang, Zhiwei Liu, Qianqian Xie, Jimin Huang, Erxue Min, and Sophia Ananiadou. Selective preference optimization via token-level reward function estimation, 2025b. URL https://arxiv.org/abs/2408.13518.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more for reasoning, 2025. URL https://arxiv.org/abs/2502.03387.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models, 2024. URL https://arxiv.org/abs/2309.12284.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models, 2023. URL https://arxiv.org/abs/2308.01825.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: Bootstrapping reasoning with reasoning, 2022. URL https://arxiv.org/abs/2203.14465.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning, 2025. URL https://arxiv.org/abs/2501.07301.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

A The Use of Large Language Models (LLMs)

In the preparation of this work, we used LLMs as auxiliary tools in a limited capacity. Specifically, LLMs assisted in drafting portions of the code and in refining the wording of certain sentences for clarity and readability. All technical content, including the design of algorithms, experimental methodology, analysis, and interpretations, was independently developed by the authors. The use of LLMs was confined to language refinement and coding suggestions, and did not influence the scientific contributions or results reported in this paper.

B Implementation Details

Code availability. Our code is available at the anonymous repository https://anonymous.4open.science/r/future-influence-D0C5.

B.1 Models and Training Configuration

We perform full-parameter supervised fine-tuning with LLaMA-Factory on Qwen3-4B-Base, Qwen3-8B-Base, Qwen3-14B-Base, and Gemma3-12B. Each model is trained separately on the two mathematical reasoning corpora described in Appendix D. The shared SFT configuration is summarized in Table 3. All compared methods use the same shared training settings within a model–corpus setting. Training lasts one epoch, and evaluation uses the checkpoint at the end of that epoch; we do not select an intermediate checkpoint according to evaluation performance.

For the recorded Gemma3 runs, the training configuration further specifies FlashAttention-2, the Liger kernel, weight decay 0.01, and left padding. Parameters not overridden in the training configuration use `seed=42`, AdamW betas $(0.9, 0.999)$, and epsilon 10^{-8}. The corresponding logs identify LLaMA-Factory version $0.9.5.dev0$.

Scoring checkpoint and fixed masks. For each setting, the scoring model is the same frozen base checkpoint used to initialize SFT. Future Influence is estimated once, before training, without updating the model parameters. The resulting supervision masks remain fixed throughout SFT. Thus, offline scoring uses the base model's predictions on the demonstrated trajectories, rather than predictions from an already fine-tuned checkpoint. Scoring and training are performed separately for the two corpora.

Table 3: Shared SFT training configuration.

Setting	Value
Framework	LLaMA-Factory ($0.9.5.dev0$ for the recorded Gemma3 runs)
Hardware	$8\times$ NVIDIA H100 GPUs
Fine-tuning strategy	Full-parameter fine-tuning
Optimizer	AdamW (`adamw_torch`)
Training precision	BF16
Sequence packing	Disabled
Learning rate	5×10^{-6}
Learning-rate schedule	Cosine decay after warmup
Warmup fraction	5%
Per-device batch size	1
Gradient accumulation steps	16
Global batch size	$1 \times 8 \times 16 = 128$
Training epochs	1
Maximum sequence length	16,384 tokens
Checkpoint selection	Final checkpoint after one epoch

B.2 Loss Normalization and Supervised Targets

The masked objective in Section 2 specifies which response targets contribute to training. Here we give the reduction used to combine their losses. For a batch of B examples, let $\mathcal{R}_b$ be the response-target index set of example b, and let $w_{b,j} \in \{0, 1\}$ indicate whether the direct loss of target j is retained. Define the number of supervised response targets in that example as

$$Z_b = \sum_{j \in \mathcal{R}_b} w_{b,j}. \tag{11}$$

We first average over the supervised targets within each example, and then average over examples:

$$\mathcal{L}_{\text{batch}}(\theta) = \frac{1}{B} \sum_{b=1}^{B} \frac{1}{Z_b} \sum_{j \in \mathcal{R}_b} w_{b,j} \left[-\log p_\theta(y_{b,j} \mid x_b, y_{b,<j}) \right]. \tag{12}$$

The same reduction is used for all compared methods. Standard SFT sets all response-target weights to one. A masking method changes the retained target set, but not the input trajectory. Consequently, a target whose direct loss is ignored can still influence the predictions of later supervised targets through the teacher-forced context. Prompt and padding positions are not response targets.

The denominator in Equation 12 counts supervised targets, not all response-content tokens. This is an example-wise mean followed by a batch mean, rather than a single token-weighted mean across the batch. EOS supervision is retained in every method, and no training example has an empty supervised-target set. This training reduction is distinct from the full-response loss used for offline gate-gradient scoring: scoring measures contextual effects on the demonstrated targets, whereas training averages only the direct target losses retained by the selected method.

B.3 Gate Placement and Information Flow

Value-path intervention. Gates are applied to the self-attention value path in every Transformer layer. Let $v_i^{(\ell)}$ denote the value vector for response token y_i at layer ℓ. We directly multiply this vector by a scalar gate:

$$\widetilde{v}_i^{(\ell)} = g_i v_i^{(\ell)}. \tag{13}$$

The same scalar g_i is shared across all Transformer layers for token i, and Qwen3 and Gemma3 use this same parameterization. Differentiating with respect to g_i aggregates the contributions of its uses across layers into a single token-level estimate. A gate value of one corresponds to the original information flow; reducing it attenuates the information carried by the corresponding response token through the value path. This intervention changes how the model uses the demonstrated context without deleting, replacing, or reordering its tokens. The input token identities, their positions, and the demonstrated targets are held fixed, while the internal representations resulting from the intervention may change.

Causal alignment. The probability assigned to y_i is computed from x and $y_{<i}$, before the model reads y_i as an input token. Intervening on the information carried by y_i therefore cannot change the prediction of y_i or any earlier target. It can affect the prediction of y_{i+1} and subsequent targets. This next-token alignment is why the full-response loss difference isolates the same effect as the loss difference restricted to $\mathcal{F}(i)$; it does not imply that the hidden representation at input position i must remain unchanged.

Content tokens and end markers. We distinguish the content-token set $\mathcal{C}$, on which directions are estimated, from the response-target set $\mathcal{R}$, on which response losses are computed. The latter includes the response end marker. EOS is not selected as a content token and retains direct supervision under every method. It also provides a future prediction target for the last response-content token. A content-token direction is attached to the loss for predicting that same token, not the loss for predicting its successor, even though the latter can contribute to its contextual influence. Qwen3 uses its official chat template. For Gemma3, the LLaMA-Factory `gemma3` template appends `<end_of_turn>` to the assistant response and treats it as the replaced EOS. Prompt, system, and user-template positions are ignored by the SFT objective; assistant content and its terminal end marker are supervised. No additional EOS label is appended for this template.

Sign convention. For our method, content targets with $s_i < 0$ have their direct loss masked, while those with $s_i \geq 0$ remain supervised. Exact zeros are assigned to the nonnegative side. The reverse comparison uses the complementary content mask, retaining only $s_i < 0$; EOS remains supervised in both cases. The same nonnegative-versus-negative convention is used when comparing estimated and directly measured directions in Appendix F.

B.4 Offline Scoring and Training Cost

Table 4 separates the three computational stages. Gate-gradient scoring produces all content-token directions in one forward and one backward pass per sequence. Because the masks are precomputed, masked SFT does not repeat this scoring operation at each optimization step. Direct intervention, in contrast, requires a separate forward evaluation for each intervened token and is used only for the diagnostic comparison. The number of these additional forward evaluations grows linearly with the number of response-content tokens. These are algorithmic pass counts, not measured wall-clock speedups.

Table 4: Computation stages.

Step	Algorithmic cost	Performed during SFT?
Future Influence scoring	One forward and one backward pass per sequence	No; offline once
Masked SFT	Standard SFT loop with ignored labels	Yes
Direct intervention validation	Separate intervention and loss computation per token	No; diagnostic only

C Comparison Methods

C.1 Internal Masking Comparisons

All internal masking comparisons keep the demonstrated input sequence intact and retain EOS supervision. Their selection rules operate on response-content tokens. For a given example, let

$$K = |\{i \in \mathcal{C} : s_i < 0\}| \tag{14}$$

be the number of content targets masked by our method. The comparisons below distinguish the number of supervised targets from the rule used to select them.

Standard SFT. SFT retains the direct prediction losses of all response targets. It uses the same complete trajectories as the selective methods, but without a content-token supervision mask.

Random masking. For each example, Random selects K content tokens at random and masks their direct losses. This matches our method's number of masked tokens separately for every sequence, rather than only matching a corpus-level retention percentage. The selected input tokens remain available as context.

Low-NLL masking. Low-NLL computes each content token's negative log-likelihood during offline scoring with the corresponding model's initialization checkpoint:

$$\ell_i = -\log p_{\theta_{\text{score}}}(y_i \mid x, y_{<i}). \tag{15}$$

It masks the K content targets with the lowest NLL in each example. Thus, the supervision budget matches our method, while the selection rule favors withholding targets that the scoring model already predicts relatively easily. This comparison uses token predictability to select the masked set; it does not match the NLL distribution of the tokens selected by Future Influence.

Reverse masking. Reverse uses exactly the same signed estimate as our method, but masks nonnegative content targets, including exact zeros, and retains negative content targets. Its content mask is

$$m_i^{\text{rev}} = \mathbb{1}\{s_i < 0\} = 1 - m_i, \qquad i \in \mathcal{C}. \tag{16}$$

EOS is excluded from this reversal and remains supervised. We do not additionally match the reverse mask's token count to K: it masks $|\mathcal{C}| - K$ content targets, whereas our method masks K. The comparison changes which side of the same signed criterion is supervised.

Table 5 summarizes the selection rules and budget relationships. All methods use the loss reduction in Appendix B.2; the corresponding performance results are reported in the main results table.

Table 5: Content-target selection rules for the internal comparisons. K is determined separately for each example by our method. All methods retain EOS supervision.

Method	Masked content targets	Number masked per example
SFT	None	0
Ours	Negative directions	K
Random	Randomly selected	K
Low-NLL	Lowest NLL	K
Reverse	Nonnegative directions	$\lvert\mathcal{C}\rvert - K$

C.2 Published Token-Level Supervision Baselines

We preserve the supervision rule and selection granularity of each published method while matching the overall response-content supervision budget when the method selects or masks targets. All published baselines use the same training trajectories, final-model initialization, shared SFT configuration, and evaluation protocol as our method.

DFT. We use Dynamic Fine-Tuning (DFT) (**?**) as a continuous token-loss reweighting baseline. During training, for each supervised response target, DFT computes its ordinary token negative log-likelihood ℓ_i and multiplies it by the detached target probability $p_i = \exp(-\ell_i)$. The probability is detached from the computation graph, so the gradient is taken only through the token loss. DFT retains all response targets, including EOS, and does not mask tokens or participate in the retention-budget matching used for target-selection baselines.

AlphaToken. **?** evaluate response tokens through path-aware gradient alignment. Their valuation combines the token's direct update path with the causal influence of the token on later autoregressive predictions, and incorporates both target adaptation and capability retention through a retention proxy. The causal path is approximated through value propagation. In our comparison, AlphaToken dynamically computes its token valuation during training and retains the top-ρ response-content targets at its prescribed batch-level granularity, where ρ is set to the overall content-token retention rate observed for our method in the same model–corpus setting. Unselected tokens remain in the input context and only their direct losses are masked; EOS is always retained and excluded from the supervision budget.

Token Cleaning. We use the Fixed-Model variant of Token Cleaning (Pang et al., 2025). For each model–corpus setting, we use the initial base checkpoint of SFT as the base model and randomly sample 20% of the trajectories from the same training corpus with a fixed random seed. Starting from that checkpoint, we warm up for one epoch to obtain a fixed reference model. We compute the negative log-likelihood of every response-content token under both models, use their difference as the token score, and apply the original method's global ranking over the entire training corpus. The total number of retained content tokens is matched exactly to the total retained by our method in the same setting. Scoring and ranking are performed once, and the resulting mask remains fixed throughout training. The final model is reinitialized from the original base checkpoint and trained with the shared SFT configuration. EOS is always retained and excluded from the supervision budget.

ssToken. For ssToken (Qin et al., 2025), we fix the SFT initialization checkpoint as the historical model and cache its token-level losses; no EMA update is used. During training, the current model recomputes token-level losses for each batch, and we extract, from the last Transformer decoder layer, the attention mass from each response token to prompt tokens, averaged over attention heads. We compute retrospective excess loss as historical-model loss minus current-model loss, min–max normalize it within each example, and combine it with the attention score using equal weights ($\gamma = 0.5$). ssToken dynamically retains the top-ρ content targets within each example, where ρ is set to the overall content-token retention rate of our method in the same model–corpus setting. Unselected tokens remain in the input context and only their direct losses are masked; EOS is always retained.

These methods therefore use different supervision rules: DFT continuously reweights all token losses by detached target probabilities; AlphaToken uses path-aware gradient alignment; Token Cleaning uses a one-time global ranking of loss differences; and ssToken dynamically combines retrospective loss with attention. Future Influence instead estimates the direction of a token's contextual effect on a fixed demonstrated continuation and uses that direction to select its direct supervision.

Matched overall retention budget. Our retention rate is determined by the observed fraction of nonnegative content-token directions rather than specified in advance. This resulting overall rate is used to configure the target-selection baselines: Token Cleaning matches the total number of retained content tokens exactly, while AlphaToken and ssToken use the same overall rate at their respective selection granularities. DFT retains all response targets and therefore does not use this masking budget. The methods therefore share the overall budget where target selection applies, while DFT provides a separate loss-reweighting comparison.

D Data Construction

D.1 Source Reasoning Corpora

OpenR1-Math-220k. OpenR1-Math-220k is a mathematical reasoning dataset released by the Open R1 project (Open R1, 2025). Its problems are drawn from NuminaMath 1.5, and its reasoning solutions are generated by DeepSeek-R1. Each example pairs a mathematical problem with a complete reasoning response, preserving the demonstrated process from problem analysis and intermediate derivations to the final answer for supervised fine-tuning on long reasoning trajectories.

Dataset construction generates multiple candidate solutions for each problem and verifies their final answers. The release primarily uses Math-Verify to check agreement with reference answers, together with model-based assessment for a subset of examples. Consequently, the candidate trajectories associated with a problem need not all have correct final answers. We select individual problem–response trajectories rather than entire groups of solutions for a problem, using complete reasoning responses with correct final answers to construct our training subset.

OpenMathReasoning. OpenMathReasoning is a large-scale mathematical reasoning corpus released by NVIDIA as part of its AIMO-2 solution (Moshkov et al., 2025). Its problems are primarily collected from the Art of Problem Solving (AoPS) community and processed into structured problem statements. Models including DeepSeek-R1 and QwQ-32B then generate reasoning solutions. The corpus contains approximately 3.2 million long chain-of-thought solutions, providing a large collection of complete reasoning demonstrations for mathematical model training.

In addition to conventional CoT solutions, the original release includes tool-integrated reasoning and solution-selection data. We use its problem–CoT response trajectories, with the problem as input and the complete reasoning response as the demonstration, without mixing in the other training formats. As with OpenR1-Math-220k, we select candidate trajectories by final-answer correctness before applying length filtering and random sampling.

D.2 Training-Subset Construction

We apply the same subset construction procedure independently to each corpus:

1. **Answer filtering.** Retain reasoning trajectories with correct final answers, selecting at the level of individual trajectories rather than problems.
2. **Length filtering.** Keep sequences shorter than 16k tokens from the answer-correct pool.
3. **Random sampling.** Randomly sample $100{,}000$ trajectories from each resulting candidate pool to form the corresponding training subset.

This produces two equally sized training corpora from different sources. OpenR1-Math-220k and OpenMathReasoning are always used independently: each model is trained separately on each corpus, rather than on their concatenation. All compared methods use the same sampled trajectories within a corpus.

Each training example retains the complete problem and demonstrated response. Answer filtering, length filtering, and random sampling determine the shared trajectory collection; the methods then construct their supervision labels on these trajectories. Standard SFT supervises all response targets, whereas token-selection methods mask the direct prediction losses of selected targets while using the same complete input sequences. Training-data construction and token-level supervision selection are thus separate stages.

E Evaluation Protocol

E.1 Mathematical Reasoning Benchmarks

We evaluate on six competition-level mathematical benchmarks spanning several years and competition formats. Each benchmark is scored by final-answer correctness, and results are reported separately rather than pooled across problems from different competitions.

AIME 2024, 2025, and 2026. The American Invitational Mathematics Examination (AIME) is an invitational mathematics competition for high-school students. Each examination contains 15 problems with integer answers between 0 and 999. An annual pair of AIME I and AIME II examinations contains 30 problems. The three yearly evaluations cover algebra, number theory, geometry, and combinatorics through problems requiring multi-step reasoning. We report the three years separately so that performance on one annual problem set does not obscure performance on another.

AMC 2023. We use the 40-problem AMC23 test snapshot distributed with Evalchemy, whose upstream dataset is `zwhe99/amc23`. The snapshot contains integer-answer problems drawn from the 2023 AMC 12A and 12B examinations. Our asset-preparation script maps each source record's identifier, question, and answer directly to the corresponding `id`, `problem`, and `answer` fields; it does not introduce multiple-choice options, letter mappings, or additional text cleaning. Final answers are evaluated with the integer extractor described below.

HMMT February 2026 and November 2025. The Harvard–MIT Mathematics Tournament (HMMT) runs distinct February and November competitions. The February competition includes subject-focused problems in algebra and number theory, geometry, and combinatorics; the November competition includes general and theme-based rounds. We evaluate final-answer problem sets from the two editions separately. The benchmark problem collections should be distinguished from a single complete individual round of the original tournament. MathArena provides curated competition problems and reference answers for AIME and HMMT evaluation (Balunović et al., 2025).

E.2 Sampling and Mathematical Metrics

All mathematical benchmarks use the same zero-shot message-level prompt:

```
system:  Please reason step by step, and put your final
answer within \boxed{}.
user:  {PROBLEM}
```

Here `{PROBLEM}` is replaced by the original problem statement. Qwen3 checkpoints serialize these messages with the official Qwen3 chat template. Gemma3 checkpoints use the LLaMA-Factory `gemma3` template: the system instruction is merged into the first user turn, enclosed by `<bos><start_of_turn>user` and `<end_of_turn>`, and followed by `<start_of_turn>model`. The benchmarks do not add task-specific system instructions or few-shot demonstrations.

For every competition problem, the evaluation directly generates 32 stochastic rollouts with temperature 0.6 and top-p 0.95, allowing at most 32k new tokens per rollout. There is no post-hoc grouping of a larger rollout set. Failed or truncated attempts remain in the denominator and are counted as incorrect.

Table 6: Generation settings for mathematical evaluation. These settings describe mathematical rollouts, not the candidate-likelihood evaluations for TruthfulQA and ARC.

Setting	Value
Prompt	Shared zero-shot reasoning prompt
Temperature	0.6
Top-p	0.95
Maximum new tokens	32k
Rollouts per problem	32
Failed attempts	Counted as incorrect
Primary metric	Directly observed pass@32
Supplementary metric	avg@1

For a benchmark with N problems, let $c_i \in \{0, \ldots, 32\}$ denote the number of correct rollouts for problem i. The primary metric is

$$\text{pass@32} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\{c_i > 0\}. \tag{17}$$

This is the observed fraction of problems solved at least once among the 32 attempts. It is not inferred from single-sample accuracy and does not use majority voting.

The supplementary single-sample metric is

$$\text{avg@1} = \frac{1}{N} \sum_{i=1}^{N} \frac{c_i}{32}. \tag{18}$$

Because every problem has the same sampling budget, this is also the fraction of correct attempts among all $32N$ attempts for that benchmark. It is computed within each benchmark, not across the six benchmark scores. The Mean avg@1 shown in Figure 2 is a separate aggregate: it is the unweighted mean of the six per-benchmark avg@1 values. Hence each benchmark contributes equally to that figure, regardless of its number of problems. Tables and plots report these accuracies as percentages.

Answer extraction and matching. For AIME 2024–2026, the evaluator searches generated text for a boxed or explicitly stated final answer and otherwise falls back to the last valid integer in $[0, 999]$. AMC23 uses the same extraction order without the 0–999 restriction, while limiting extracted integer strings to at most four digits. HMMT February 2026 and November 2025 use `math_verify` to parse and verify mathematical equivalence, with a 30-second timeout per comparison. Truncated and otherwise unverifiable generations are scored as incorrect. The evaluation checkout is recorded at revision `cefdedda7b3c5ee7f7fc87a5bbba59d4dc756108`; each saved summary additionally records its protocol version.

E.3 Non-Mathematical Tasks and Metrics

We additionally evaluate instruction following, truthfulness, and science question answering using the official task prompts. The three benchmark families produce five reported metrics: IFEval strict accuracy, TruthfulQA MC1 and MC2, and ARC-Challenge accuracy and length-normalized accuracy. These evaluations characterize behavior outside the mathematical training domain; they are reported separately from the primary mathematical results.

IFEval. IFEval (Zhou et al., 2023) tests instruction following with automatically verifiable constraints. The original benchmark contains 541 prompts covering 25 types of verifiable instructions, such as length constraints, required keywords, and output-format requirements. A prompt can contain more than one constraint. We greedily decode one response per prompt with a maximum context and completion request of $16{,}384$ tokens, remove only a complete `<think>...</think>` block when present, and apply the official strict and loose rule checkers from lm-evaluation-harness. The reported IFEval strict score is `prompt_level_strict_acc`: a prompt is counted as correct only if all of its instructions pass their strict checks. The instruction-level strict score is also computed but is not reported in the main table.

TruthfulQA. TruthfulQA (Lin et al., 2021) consists of 817 questions across 38 topic categories and is designed to expose false answers arising from common misconceptions. We report its multiple-choice metrics rather than free-generation truthfulness ratings. MC1 measures whether the designated best correct answer receives a higher score than the incorrect candidates. MC2 measures the normalized probability mass assigned to all correct candidates. Writing $\ell(a \mid q)$ for the conditional log-likelihood of candidate answer a, and A_q^+ and A_q^- for the correct and incorrect candidate sets, the per-question MC2 score is

$$\mathrm{MC2}(q) = \frac{\sum_{a \in A_q^+} \exp \ell(a \mid q)}{\sum_{a \in A_q^+ \cup A_q^-} \exp \ell(a \mid q)}. \tag{19}$$

The benchmark score averages this quantity over questions. MC2 therefore credits probability assigned to multiple correct answers, whereas MC1 uses the designated best answer for its single-choice comparison.

ARC-Challenge. ARC (Clark et al., 2018) contains science questions intended to assess knowledge and reasoning. Its Challenge subset comprises questions not correctly answered by the information-retrieval and word-co-occurrence baselines used in constructing the dataset. We use the `ARC-Challenge` test split from `allenai/ai2_arc`. For each question, the chat-level user prompt is `Question: {question}` `textbackslash nAnswer:`, and each original choice string is scored as a teacher-forced continuation. The reported `acc` selects the choice with the greatest total conditional log-likelihood. The supplementary `acc_norm` instead divides total log-likelihood by the number of Python string characters in the original choice text before selecting the answer; its normalization unit is therefore characters, not tokens. Both metrics average correctness of the selected choices over questions.

Task prompts and evaluation modes. IFEval supplies the instruction-bearing prompt for each example, so the task input includes the constraints that the verifier will check. TruthfulQA and ARC instead evaluate the conditional likelihoods of candidate answers under their task prompts. They do not use the 32-rollout mathematical protocol. ARC uses the explicit question-and-answer prompt described above. All tasks use the model-family formatting used in training: the official Qwen3 chat template for Qwen3 and the LLaMA-Factory `gemma3` template for Gemma3. No method-specific prompt is introduced for masked models.

General-performance summary. The *Gen.* column in the main table is the equal-weight arithmetic mean of the five reported non-mathematical percentages. It provides a compact supplementary summary; the constituent scores remain available in Table 7. Improvements in this aggregate need not imply improvements in every constituent metric.

F Validation of Gate-Gradient Directions

F.1 Samples and Interventions

The validation settings are those listed in Table 2: Qwen3-14B-Base and Gemma3-12B on OpenR1-Math-220k and OpenMathReasoning. We randomly draw $1{,}000$ examples from each dataset. For each example, all response-content tokens contribute to the comparison. A single forward and backward pass first produces the estimates s_i at the original gate setting. Direct interventions are then evaluated individually, so that each comparison concerns one token rather than a simultaneous perturbation of a group of tokens.

We distinguish three diagnostic interventions:

- **Small attenuation ($g_i = 0.99$).** Only the selected token's gate is reduced from one to 0.99. This probes the direction of the loss change close to the point at which the gradient is evaluated.
- **Full suppression ($g_i = 0$).** The selected gate is set to zero under the same gate parameterization. The input sequence and demonstrated targets remain unchanged, but the step is no longer a small local perturbation.
- **Token removal.** The selected input token is physically removed and the continuation shifts forward. This is an additional diagnostic: unlike the two gate interventions, it changes the sequence and its positional structure rather than only attenuating information flow.

For the gate interventions, all other gates remain at one, and the future-target set is held fixed between the original and intervened evaluations. Each intervention type is evaluated separately. EOS is a future prediction target but not a content token selected for intervention.

F.2 Token-Pooled Agreement and Exact Zeros

Agreement compares the nonnegative-versus-negative classification of the estimate with that of the directly measured loss change. Define

$$h(z) = \mathbb{1}\{z \geq 0\}. \tag{20}$$

An exact zero is assigned to the nonnegative side, for both the gradient estimate and the direct intervention. Although exact zeros are rare, this convention makes the comparison well defined without discarding them or introducing a tolerance threshold.

For a fixed model, dataset, and intervention type, let $\mathcal{C}_b$ be the content-token set of validation example b. We pool all tokens across examples and compute

$$\text{Agreement} = \frac{\sum_b \sum_{i \in \mathcal{C}_b} \mathbb{1}\{h(s_{b,i}) = h(I_{b,i})\}}{\sum_b |\mathcal{C}_b|}. \tag{21}$$

Here $I_{b,i}$ is the direct loss change for the intervention being evaluated. Each content token contributes one comparison, so longer responses contribute more comparisons than shorter ones. This differs from first computing an agreement rate for each response and then averaging responses equally. Positive and zero values agree with each other under h; either disagrees with a negative value. The definition is shared across the three diagnostic interventions.

Reported results. Table 2 reports all four model–dataset settings. Agreement ranges from 0.9932 to 0.9983 under small gate attenuation, from 0.9667 to 0.9815 when the gate is set to zero, and from 0.9425 to 0.9541 under token removal. Every setting follows the same ordering, with small attenuation giving the highest agreement, followed by zeroing the gate and then token removal. These measurements quantify agreement of the negative-versus-nonnegative classifications.

G Non-Mathematical Benchmarks

Table 7 reports IFEval (strict), TruthfulQA MC1 and MC2, and ARC-Challenge accuracy and length-normalized accuracy for every model–dataset setting and method for which results are avail-

able. These numbers are not part of the primary evidence; they indicate what changing the supervised set of mathematical reasoning tokens carries over to instruction following and factuality. The two ARC columns are the standard `acc` and `acc_norm` metrics for ARC-Challenge.

Table 7: IFEval (strict), TruthfulQA MC1 and MC2, and ARC-Challenge accuracy and length-normalized accuracy recorded in the source results. The two panels correspond to the two training datasets and include all methods for each model.

OpenR1-Math-220k						OpenMathReasoning					
Method	IFEval	MC1	MC2	ARC-C acc	ARC-C acc_norm	Method	IFEval	MC1	MC2	ARC-C acc	ARC-C acc_norm
Qwen3-4B-Base						*Qwen3-4B-Base*					
SFT	38.26	28.15	44.75	38.82	39.93	SFT	30.13	27.91	44.28	38.65	37.88
DFT	38.49	28.15	44.95	38.92	40.24	DFT	31.59	28.22	44.28	38.54	37.67
AlphaToken	38.54	28.06	44.58	38.65	39.79	AlphaToken	30.42	27.91	44.28	38.65	37.88
Cleaning	38.05	27.91	44.38	38.48	39.65	Cleaning	29.79	27.66	44.07	38.48	37.67
ssToken	38.96	28.27	44.86	38.82	39.93	ssToken	30.83	28.06	44.48	38.82	38.05
Random	40.11	28.64	46.08	38.05	38.99	Random	31.05	28.52	44.87	39.76	38.65
Low-NLL	41.96	26.93	44.40	38.99	40.19	Low-NLL	30.68	27.91	44.28	38.91	38.05
Reverse	40.34	27.66	44.84	38.12	39.36	Reverse	31.84	28.03	44.56	38.95	38.33
Ours	41.04	26.93	44.11	36.60	38.31	**Ours**	35.49	27.78	44.79	38.48	38.74
Qwen3-8B-Base						*Qwen3-8B-Base*					
SFT	30.50	27.91	43.94	34.13	37.80	SFT	23.84	27.66	43.46	34.39	38.99
DFT	32.19	27.80	43.73	34.23	37.80	DFT	25.01	27.66	43.46	34.49	38.88
AlphaToken	30.21	27.91	43.75	34.13	37.60	AlphaToken	24.17	27.71	43.54	34.56	39.10
Cleaning	29.79	27.66	43.54	33.87	37.40	Cleaning	23.54	27.41	43.33	34.28	38.85
ssToken	30.63	28.06	44.17	34.28	37.90	ssToken	24.38	27.91	43.67	34.69	39.25
Random	26.99	28.27	44.83	33.87	36.69	Random	22.55	29.38	44.66	34.90	38.31
Low-NLL	30.31	27.91	43.62	34.13	37.20	Low-NLL	27.17	27.91	43.84	35.75	39.16
Reverse	28.13	27.71	43.79	34.28	37.56	Reverse	25.09	27.97	43.56	35.28	38.97
Ours	25.69	27.05	43.17	34.56	38.05	**Ours**	26.80	26.93	42.28	36.09	39.42
Qwen3-14B-Base						*Qwen3-14B-Base*					
SFT	34.08	24.25	45.76	41.13	41.30	SFT	25.42	26.46	44.79	42.92	43.54
DFT	35.11	24.25	45.65	41.44	41.50	DFT	26.29	26.66	44.99	42.92	43.74
AlphaToken	34.21	24.42	45.21	44.79	45.00	AlphaToken	25.63	26.67	45.21	43.13	43.75
Cleaning	33.75	24.17	44.79	44.38	44.58	Cleaning	25.21	26.46	44.79	42.50	43.13
ssToken	34.38	24.58	45.63	45.21	45.42	ssToken	26.04	26.88	45.42	43.33	43.96
Random	33.09	24.85	43.27	45.82	45.56	Random	24.21	26.81	44.50	42.83	44.03
Low-NLL	33.83	25.34	43.24	45.22	46.25	Low-NLL	26.62	26.93	45.60	42.75	43.26
Reverse	33.54	25.04	43.96	44.38	44.92	Reverse	24.79	26.46	44.38	42.50	43.02
Ours	34.38	25.83	44.17	44.03	44.80	**Ours**	26.25	26.88	45.42	43.13	43.75
Gemma3-12B						*Gemma3-12B*					
SFT	26.80	25.09	40.57	38.91	39.93	SFT	22.18	25.34	40.44	38.14	37.88
DFT	27.26	25.09	40.67	39.01	39.82	DFT	22.85	25.44	40.44	37.93	38.19
AlphaToken	26.46	25.09	40.38	38.74	39.79	AlphaToken	21.88	25.34	40.31	38.14	37.88
Cleaning	26.21	24.85	40.13	38.57	39.65	Cleaning	21.46	25.21	40.06	37.97	37.67
ssToken	26.97	25.21	40.63	38.91	39.93	ssToken	22.29	25.46	40.44	38.23	38.05
Random	24.95	25.09	38.80	38.82	38.99	Random	21.26	25.46	39.91	38.23	38.05
Low-NLL	25.21	25.34	39.91	38.74	39.30	Low-NLL	21.67	25.46	40.31	38.40	38.10
Reverse	23.84	24.60	39.39	37.80	38.60	Reverse	19.78	25.34	39.67	37.63	37.30
Ours	23.48	26.32	39.46	39.33	39.90	**Ours**	19.41	25.58	40.58	38.57	38.74

H AVERAGE SINGLE-SAMPLE ACCURACY

The primary metric remains pass@32. To characterize single-sample behavior, Table 8 reports avg@1 separately for each competition-level benchmark and training dataset, where avg@1 denotes the average single-sample accuracy over all 32 rollouts used for each problem; it is not an average across benchmarks. The two metrics give the same ordering: avg@1 is highest for our method in 46 of the 48 combinations, tied in one, and below random masking in one (Qwen3-8B-Base on OpenR1-Math-220k, AIME 2026). The margins are smaller than at the larger sampling budget.

Table 8: avg@1 on six competition-level benchmarks for models trained on OpenR1-Math-220k and OpenMathReasoning. HMMT26 and HMMT25 denote HMMT February 2026 and HMMT November 2025.

	OpenR1-Math-220k						OpenMathReasoning					
Method	AIME24	AIME25	AIME26	AMC23	HMMT26	HMMT25	AIME24	AIME25	AIME26	AMC23	HMMT26	HMMT25
Qwen3-4B-Base												
SFT	22.29	23.12	19.17	62.66	12.69	10.00	18.96	20.62	18.12	54.06	14.58	10.21
DFT	22.29	23.02	19.27	62.73	12.78	9.90	19.06	20.73	18.13	54.14	14.58	10.21
AlphaToken	22.71	22.92	19.58	62.50	12.31	10.21	19.38	20.83	18.33	54.38	14.39	10.42
Cleaning	22.08	23.33	18.96	62.34	13.16	9.79	18.75	20.42	17.92	53.75	14.77	10.00
ssToken	22.50	23.12	18.75	62.66	12.88	10.00	19.17	20.42	18.12	54.06	14.77	10.21
Random	21.88	19.79	20.00	60.78	13.26	8.75	18.75	19.38	18.12	48.75	11.17	9.17
Low-NLL	22.92	21.88	18.12	62.50	14.96	9.17	20.62	22.29	16.88	54.22	14.77	11.46
Reverse	20.00	21.67	18.54	54.69	10.42	7.29	15.42	17.92	12.92	45.16	9.47	3.75
Ours	23.33	23.54	21.46	63.13	15.44	10.42	21.67	22.71	18.75	61.72	15.25	11.88
Qwen3-8B-Base												
SFT	26.25	24.58	25.83	72.19	16.86	17.08	29.17	24.17	25.62	70.78	17.80	13.33
DFT	26.46	24.58	25.94	72.27	16.95	17.19	29.27	24.06	25.63	70.70	17.90	13.44
AlphaToken	26.67	24.58	26.25	71.88	17.33	16.67	29.58	24.17	26.04	70.31	18.09	12.92
Cleaning	25.83	25.00	25.83	72.19	16.86	17.29	28.75	24.58	25.62	70.78	17.80	13.75
ssToken	26.25	24.17	25.42	72.50	16.48	17.08	29.17	23.75	25.21	71.25	17.42	13.33
Random	25.42	23.75	26.88	67.03	15.34	14.79	24.58	24.38	22.50	64.84	18.56	14.38
Low-NLL	29.38	25.83	25.83	69.84	18.94	15.21	25.83	23.54	26.46	72.34	19.13	13.96
Reverse	21.25	22.92	19.17	65.00	14.96	12.29	20.62	20.42	17.50	61.09	12.31	9.38
Ours	29.79	26.25	26.67	72.66	19.60	17.50	30.00	26.46	28.75	72.81	19.79	16.46
Qwen3-14B-Base												
SFT	41.77	31.67	41.67	79.14	23.30	29.17	41.67	31.67	40.42	79.61	23.77	25.42
DFT	41.67	31.77	41.88	79.14	23.48	29.38	41.77	31.67	40.63	79.61	23.67	25.52
AlphaToken	42.50	32.08	41.25	80.39	23.30	29.58	42.50	32.50	41.25	80.39	23.77	25.42
Cleaning	41.25	31.25	41.67	79.61	23.30	28.75	41.25	32.08	40.42	79.14	23.77	25.42
ssToken	42.08	32.50	42.50	79.14	23.77	29.17	42.92	32.92	40.83	80.86	23.77	25.21
Random	40.31	31.25	41.25	79.22	22.92	29.17	40.83	31.88	38.33	81.88	23.30	23.54
Low-NLL	45.00	32.71	42.08	82.66	24.05	29.38	44.58	34.38	40.62	82.50	23.30	26.67
Reverse	37.71	30.83	30.63	80.00	19.89	22.71	38.12	27.50	31.67	77.97	21.21	21.88
Ours	45.42	33.33	42.92	84.14	24.62	30.00	45.83	35.00	43.65	83.36	24.15	27.50
Gemma3-12B												
SFT	12.50	12.29	10.21	46.64	5.21	4.38	14.27	14.58	11.77	51.41	8.24	6.35
DFT	12.60	12.40	10.42	46.64	5.21	4.38	14.17	14.58	11.67	51.56	8.24	6.46
AlphaToken	12.50	12.40	10.21	46.56	5.21	4.58	14.38	14.79	11.77	51.25	8.24	6.35
Cleaning	12.40	12.29	10.42	46.64	5.30	4.38	14.06	14.58	11.98	51.56	8.33	6.35
ssToken	12.50	12.19	10.00	46.72	4.92	4.17	14.27	14.38	11.56	51.41	8.14	6.46
Random	12.50	12.19	10.00	46.56	5.11	4.27	12.71	13.96	10.83	49.61	5.78	5.52
Low-NLL	12.60	12.40	10.63	46.72	5.30	4.58	14.48	14.90	11.98	51.72	8.52	6.56
Reverse	11.25	11.04	8.96	44.53	4.17	3.75	12.19	12.50	10.62	46.56	5.68	5.00
Ours	12.71	12.50	10.83	46.80	5.30	5.05	14.58	15.00	12.19	52.03	8.62	6.77

I Additional Mask Statistics

I.1 Counting Unit and Corpus-Level Fractions

The mask statistics concern response-content tokens with estimated directions, not all positions in the concatenated prompt–response input. In particular, EOS is not included in the content-token denominator even though it remains a supervised response target. For a scored collection of examples, the fraction assigned a negative direction is

$$r_{\mathrm{neg}} = \frac{\sum_b \sum_{i \in \mathcal{C}_b} \mathbb{1}\{s_{b,i} < 0\}}{\sum_b |\mathcal{C}_b|}. \tag{22}$$

This token-pooled fraction is also the fraction of content targets whose direct supervision is masked by our method. The number of supervised response targets additionally includes the retained end markers, so content-token masking percentages should not be interpreted as percentages over all model input positions.

The reported statistics cover the three Qwen3 model sizes and Gemma3-12B crossed with the two training corpora, giving eight score files with $100{,}000$ examples each. Summed over model settings, the OpenR1-Math-220k files contain approximately 1.329 billion scored content tokens, and the OpenMathReasoning files contain approximately 3.340 billion. These totals count scored tokens across model settings rather than distinct textual tokens in one training set. Across the eight files, the negative fraction ranges from 50.21% to 51.83%. The Qwen3 and Gemma3 statistics use their respective model tokenizers and chat templates.

I.2 Zeros and Reverse-Mask Counts

The six Qwen3 score files contain one exact zero in total; the Gemma3 files report 0.00% at the displayed precision. Our method retains exact-zero supervision, whereas Reverse masks it. This fixes the zero boundary even though its numerical contribution to the aggregate percentages is negligible. For each example, the two methods' masked content counts sum to $|\mathcal{C}|$, but those counts need not be equal. Thus, the approximately balanced corpus-level signs do not imply an exactly matched per-example budget for Reverse. Random and Low-NLL, in contrast, explicitly match our method's masked count per example.

I.3 Position Within the Response

Figure 3 groups response-content tokens into 20 equal-width bins of normalized position. Each bin reports the fraction of its scored content tokens with negative direction. The observed within-setting range across bins is at most 3.86 percentage points for the eight reported settings. This describes the positional distribution of the selected mask; it does not assign a semantic role to any position or identify tokens as correct, redundant, or important. Normalized-position bins aggregate many absolute distances from the response end, so this coarse profile alone does not establish a token-by-token pattern at the sequence tail.