Small Vectors, Big Effects: A Mechanistic Study of RL-Induced Reasoning via Steering Vectors

Viacheslav Sinii* **Nikita Balagansky** **Yaroslav Aksenov**

Vadim Kurochkin **Daniil Laptev** **Alexey Gorbatovski**

Boris Shaposhnikov **Daniil Gavrilov**

T-Tech

Abstract

The mechanisms by which reasoning training reshapes LLMs' internal computations remain unclear. We study lightweight steering vectors inserted into the base model's residual stream and trained with a reinforcement-learning objective. These vectors match full fine-tuning performance while preserving the interpretability of small, additive interventions. Using logit-lens readouts and path-patching analyses on two models, we find that (i) the last-layer steering vector acts like a token-substitution bias concentrated on the first generated token, consistently boosting tokens such as "To" and "Step"; and (ii) the penultimate-layer vector leaves attention patterns largely intact and instead operates through the MLP and unembedding, preferentially up-weighting process words and structure symbols. We also show that steering vectors (i) transfer to other models, (ii) combine across layers when trained in isolation, and (iii) concentrate magnitude on meaningful prompt segments under adaptive token-wise scaling. Taken together, these results deepen understanding of how trained steering vectors shape computation and should inform future work in activation engineering and the study of reasoning models.

1 Introduction

Reasoning-oriented language models have recently made striking gains Jaech et al. [2024], Guo et al. [2025]. Many top systems are trained with reinforcement learning on verifiable tasks, especially mathematics, where correctness provides reliable rewards. Yet we still lack a mechanistic account of what this training changes inside the network.

Following Sinii et al. [2025], we train *steering vectors* – learned additive directions injected into the residual stream, while freezing the base model. This parameterization isolates a small set of features that can be probed, ablated, and composed with mechanistic-interpretability tools. To isolate layer-wise effects, we fit one steering vector per layer ℓ and measure its behavioral impact, then analyze in depth the layers with the clearest effects.

Our contributions are as follows. **Layer-wise isolation of RL-induced gains.** One steering vector per layer, frozen base; evaluated across two models and six math benchmarks. **Last layer behaves like first-token substitution.** The final-layer vector acts at unembedding, boosting opening tokens (e.g., "To"/"Step"); simply prefixing that token recovers $\sim$10–11 points – about three quarters of

*Corresponding author: v.siniy@t-tech.dev

Preprint.

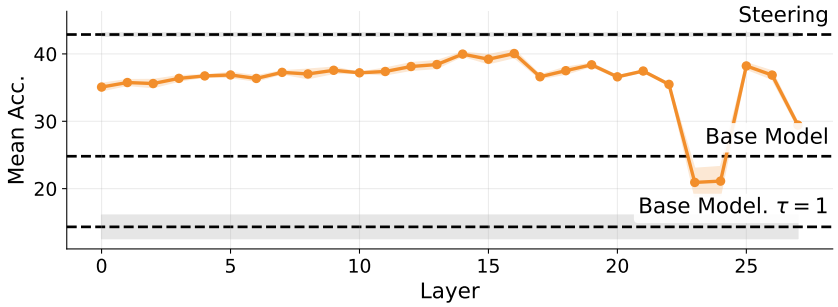


Figure 1: **Single-layer steering.** Mean accuracy on six benchmarks for Qwen2.5-Math-7B when training a single vector s_ℓ at layer ℓ with all other layers frozen. Mid-layer vectors yield the largest gains but never match all-layer steering, indicating the improvement is distributed across layers.

the explicit last-layer gain. **Penultimate-layer circuit for Qwen-2.5-Math-7B.** Skip-Attn retains over half the gain, and injecting only into V_1 recovers the full penultimate-layer effect; analytically, the value-path addition yields a linear term independent of attention, consistent with an OV+MLP mechanism plus a direct unembedding push.

2 Single-layer steering vectors

Setup. We study two base models – Qwen2.5-Math-7B [Team, 2024] and Llama3.1-8B-Instruct [Grattafiori et al., 2024]. Steering vectors are trained with the RLOO objective [Ahmadian et al., 2024] on the DeepScaleR dataset [Luo et al., 2025]. Evaluation spans six math benchmarks: AIME24/25, AMC23, MATH500 [Hendrycks et al., 2021], MinervaMath [Lewkowycz et al., 2022], and OlympiadBench [He et al., 2024]. We report the mean score across these benchmarks. See Section B for further details.

Result. For each layer ℓ, we train a single steering vector s_ℓ while freezing all others. Figure 1 reports per-layer results for Qwen2.5-Math-7B (see Section C for LLaMa3.1-8B-It), compared with (i) all-layer steering, (ii) the base model with greedy decoding, and (iii) the base model sampled at $\tau = 1.0$ (the training initialization). Most layers improve over the initialization, but none matches all-layer steering; under greedy decoding, several do (Section D), suggesting that single-layer vectors target the right mechanisms yet cannot on their own sufficiently reduce the next-token distribution's entropy. In Qwen2.5-Math-7B, s_{23} and s_{24} underperform their neighboring layers; we trace the issue to vectors passing through the input layer-norm in layer 25 (Section E).

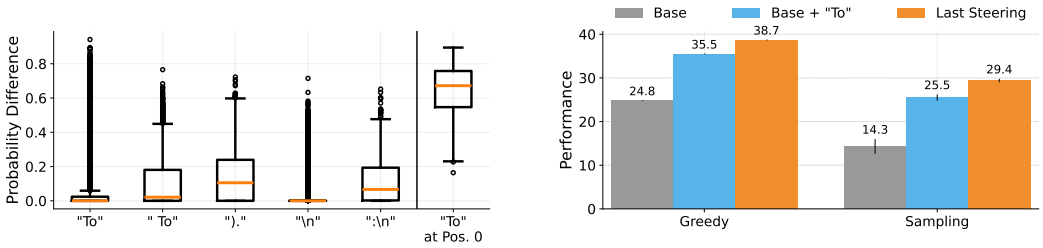
The value of single-layer steering is the simplified interpretability: each vector's effect is isolated, avoiding cross-layer interactions. We build on this in the next section.

3 Last Layer – Token Substitution

Training only the last-layer vector s_{27} closes over 50% of the gap between the base model and all-layer steering, indicating a strong task signal. With no subsequent layers to process it, s_{27} acts at unembedding without altering hidden states, effectively substituting tokens by boosting the logit of those it aligns with. We read out these preferences via a `logit-lens` projection [nostalgebraist, 2020], multiplying s_{27} by the unembedding matrix (omitting the pre-unembed layer norm); the top token is *"To"* (score 42.5; cosine 0.37) (see Section F for other top-10 tokens). Though the vector is added unconditionally, because softmax is nonlinear, effects vary by position, so we estimate the induced probability differences on 256 DeepScaleR prompts:

$$\Delta P_i = P(V_i \mid x_{:t}; \theta, s_{27}) - P(V_i \mid x_{:t}; \theta).$$

Grouping by token, the largest increases are for *"To"* and *" To"*, concentrated at the first generated token (Figure 2a). To test first-token steering directly, we simply append *"To"* to each prompt and evaluate the *base* model: accuracy rises by 10-11 points under both greedy and sampling – about 75% of the gain from s_{27} (Figure 2b).



(a) Qwen2.5-Math-7B: Distribution of token-level probability change ΔP induced by the last-layer vector over 256 DeepScaleR prompts. Five tokens with the largest maxima are shown and a separate distribution for "*To*" at the first generation position.

(b) Prepending "*To*" to each prompt raises base-model accuracy by 10–11 points under both greedy and sampling, capturing about 75% of the gain from the explicit last-layer vector.

Figure 2: **Last-layer analysis.** Left: the last-layer vector mainly boosts the initial token "*To*". Right: prefixing that token reproduces most of the observed performance gain.

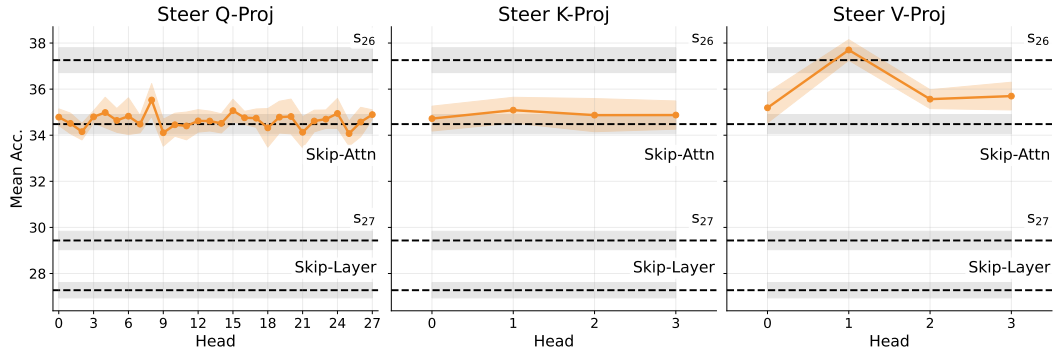


Figure 3: **Penultimate-layer steering in Qwen2.5-Math-7B.** Mean accuracy when injecting s_{26} into a single projection of the final block: Q (left), K (center), V (right). Placing s_{26} only in V_1 closes the gap between *Skip-Attn* and s_{26}, indicating the effect is carried by the $V_1 \to W^O$ path and is largely independent of Q/K and attention weights.

The last-layer vector in LLaMa3.1-8B-It has a much smaller impact (Section C), suggesting token substitution is less effective. Nonetheless, s_{31} again concentrates its influence on the first generated token and preferentially promotes *"Step"* (Section G).

4 Penultimate Layer – Circuit

Steering the penultimate layer s_{26} yields a larger accuracy gain than steering the last layer, and remains tractable to analyze because the modified activations traverse only one remaining block. Here we identify which parts of that block convert the steering signal into performance.

For residual input X, the block computes $Y = X + \mathrm{MHA}(\mathrm{LN}(X))$ and $Z = Y + \mathrm{MLP}(\mathrm{LN}(Y))$, with heads $H_i(U) = \mathrm{Softmax}(UW_i^Q(UW_i^K)^\top/\sqrt{d_k})\,UW_i^V$ concatenated and mixed by W^O, and an MLP $f(UW_1 + b_1)W_2 + b_2$. We assess the contribution of each submodule by inserting or omitting s_{l-1} at specific locations and measuring mean accuracy: **Full steering** $X \leftarrow X + s_{l-1}$; **Skip-Layer** $Z \leftarrow Z + s_{l-1}$; **Skip-Attn** $Y \leftarrow Y + s_{l-1}$; **Steer-Q/K/V-Proj** for a head i, $(UW_i^{Q/K/V}) \mapsto (U + s_{l-1})W_i^{Q/K/V}$.

Figure 3 gives three takeaways: (i) *Skip-Layer* reduces accuracy relative to passing s_{26} through the block, but the drop is small compared with using s_{27}; thus s_{26} still helps via a direct push on the unembedding, with the remainder coming from in-block processing; (ii) *Skip-Attn* preserves over half of the s_{26} gain, pointing to the MLP as the main contributor; (iii) patching any single Q or K, or a V_j with $j \neq 1$, has little effect, whereas placing s_{26} only in V_1 closes the gap to the full s_{26} result.

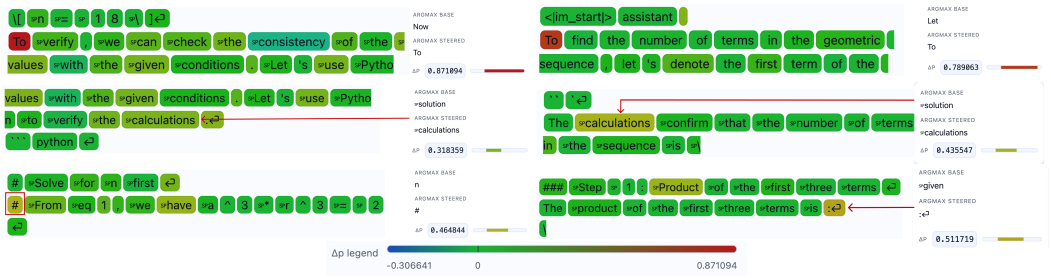


Figure 4: **Case study (Qwen2.5-Math-7B).** Token-level probability shifts (Δp) induced by *penultimate-layer* steering. Three patterns emerge: **row 1** amplify the paragraph-initial token "*To*"; **row 2** suppress "*solution*" in favor of "*calculations*"; **row 3** favor structural tokens that start Python code comments, and newlines – rather than continuing the current sentence.

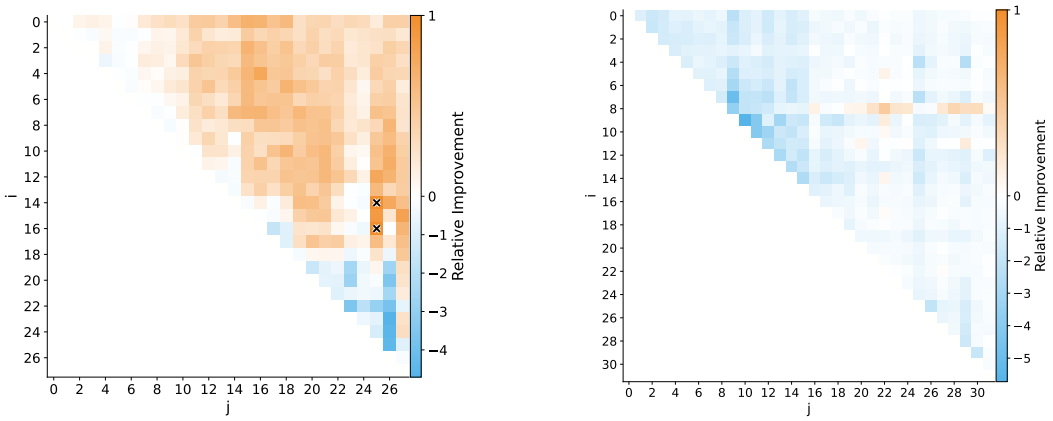


Figure 5: **Composable steering.** The normalized gain in mean accuracy when pairing vectors s_i and s_j with $i < j$. Crosses mark pairs reaching $\geq 99\%$ of all-layer. Qwen2.5-Math-7B often benefits, with two near–all-layer pairs; LLaMa3.1-8B-It is mostly neutral or interfering.

Viewed through the QK/OV-circuit lens [Elhage et al., 2021], token–token interaction (QK) is unchanged and the effect travels through the OV path – controlling what is written when a token is attended. Moreover, because $s_{l-1} W_1^V W_1^O$ enters the residual regardless of attention weights (Section H), this is equivalent to adding the projected vector just before the MLP, i.e., skipping attention. Indeed, a vector trained directly on the post-attention residual reaches 38.8 ± 0.6 mean accuracy, matching s_{26}. Overall, the penultimate vector in Qwen2.5-Math-7B acts via two routes: a direct effect on the unembedding and an interaction with the MLP. Section I contains the LLaMa3.1-8B-It study.

Figure 4 shows how adding the steering vector to the post-attention residual stream shifts token probabilities. Beyond boosting the probability of the paragraph-initial token *To*, it promotes process words – e.g., replacing "solution" with "calculations," possibly to deter premature endings. It also favors structural tokens such as Python comment markers and newlines, which often precede math blocks and may support in-code reasoning.

5 Steering Vectors are Composable

We test whether depth-specific vectors combine without conflict by evaluating all pairs (s_i, s_j) with $i < j$. Figure 5 reports the normalized gain

$$\mathrm{norm}(s_i, s_j) = \frac{\mathrm{Acc}(s_i, s_j) - \max\{\mathrm{Acc}(s_i),\ \mathrm{Acc}(s_j)\}}{\mathrm{Acc}(\mathbb{S}) - \max\{\mathrm{Acc}(s_i),\ \mathrm{Acc}(s_j)\}},$$

which compares the pair to the better single vector: 0 means no improvement, 1 matches the all-layer score $\mathbb{S}$, and < 0 indicates interference. Exact accuracies are in Section J.

Adjacent pairs (near the diagonal) often interfere, while wider gaps add constructively. Notably, s_{25} paired with s_{16} or s_{14} nearly matches the all-layer 42.9%, though each alone plateaus around 40%. The composition in LLaMa3.1-8B-It is weaker: many pairs are neutral or harmful. The modest gains concentrate when late layers are paired with mid-depth layer 8, but none reach the all-layer result.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*, 2024.

Jan Betley, Xuchan Bao, Martín Soto, Anna Sztyber-Betley, James Chua, and Owain Evans. Tell me about yourself: Llms are aware of their learned behaviors. *arXiv preprint arXiv:2501.11120*, 2025.

Yuanpu Cao, Tianrong Zhang, Bochuan Cao, Ziyi Yin, Lu Lin, Fenglong Ma, and Jinghui Chen. Personalized steering of large language models: Versatile steering vectors through bi-directional preference optimization. *Advances in Neural Information Processing Systems*, 37:49519–49551, 2024.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

Josh Engels, Neel Nanda, and Senthooran Rajamanoharan. Interim research report: Mechanisms of awareness. *AI Alignment Forum*, 2025. `https://www.alignmentforum.org/posts/m8WKfNxp9eDLRkCk9/interim-research-report-mechanisms-of-awareness`.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Sheng Liu, Haotian Ye, Lei Xing, and James Zou. In-context vectors: Making in context learning more effective and controllable through latent space steering. *arXiv preprint arXiv:2311.06668*, 2023.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. `https://pretty-radio-b75.notion.site/`

DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a
2025. Notion Blog.

Andrew Mack and Alex Turner. Mechanistically eliciting latent behaviors in language models. *AI Alignment Forum*, 2024. https://www.alignmentforum.org/posts/ioPnHKFyy4Cw2Gr2x/mechanistically-eliciting-latent-behaviors-in-language-1.

nostalgebraist. interpreting gpt: the logit lens, 2020. https://www.alignmentforum.org/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*, 2023.

Viacheslav Sinii, Alexey Gorbatovski, Artem Cherepanov, Boris Shaposhnikov, Nikita Balagansky, and Daniil Gavrilov. Steering llm reasoning through bias-only adaptation. *arXiv preprint arXiv:2505.18706*, 2025.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Constantin Venhoff, Iván Arcuschin, Philip Torr, Arthur Conmy, and Neel Nanda. Understanding reasoning in thinking language models via steering vectors. *arXiv preprint arXiv:2506.18167*, 2025.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

Jake Ward, Chuqiao Lin, Constantin Venhoff, and Neel Nanda. Reasoning-finetuning repurposes latent representations in base models. *arXiv preprint arXiv:2507.12638*, 2025.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*, 2025.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A Related Work

Steering vectors are small additive perturbations to the residual stream that modulate model behavior. They are widely viewed as *feature amplifiers* – strengthening existing computations rather than introducing new mechanisms – and have been used to toggle or amplify *reasoning-like* behaviors [Venhoff et al., 2025, Ward et al., 2025]. A common way to obtain them is *contrastive extraction* from activation pairs (e.g., positive vs. negative sentiment) [Turner et al., 2023, Panickssery et al., 2023, Liu et al., 2023, Zou et al., 2023]. Beyond extraction, steering directions can also be *trained*: optimized with preference data for controllable generation [Cao et al., 2024], or learned as simple additive vectors that surface latent behaviors such as step-by-step reasoning or self-reflection [Mack and Turner, 2024, Engels et al., 2025, Betley et al., 2025].

In this work, we interpret steering vectors trained with GRPO-like objective using standard tools from mechanistic interpretability – `logit-lens` to read out token-level effects [nostalgebraist, 2020], *path patching* to localize circuits [Wang et al., 2022], and circuit-style analyses in the QK/OV framework [Elhage et al., 2021].

B Setup details

Training uses the sampling temperature $\tau = 1.0$, a 4K context window for Qwen2.5-Math-7B, and 8K for Llama3.1-8B-Instruct. Rewards are assigned with `Math-Verify`[1]. For MATH500, MinervaMath, and OlympiadBench we report PASS@1; for AIME24/25 and AMC23 we report AVG@32 due to their smaller sizes. Base models decode greedily; trained models decode with sampling at $\tau = 1.0$ following Zeng et al. [2025]. Evaluation context length is 4K and 32K for Qwen2.5-Math-7B and other models respectively. All metrics are averaged over three evaluation seeds.

C Single-layer LLaMa3.1-8B-It

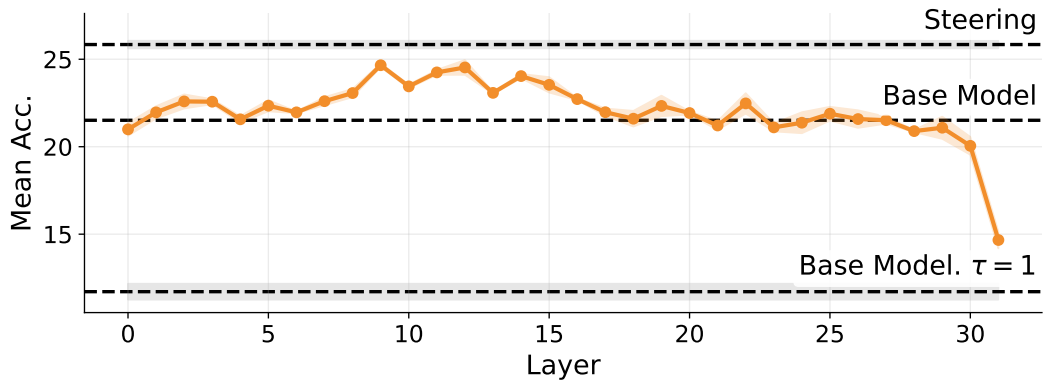


Figure 6: **Single-layer steering.** Mean accuracy on six benchmarks for Qwen2.5-Math-7B when training a single vector s_ℓ at layer ℓ with all other layers frozen. Vectors from layers $8 - 15$ yield the largest gains but never match all-layer steering, indicating the improvement is distributed across layers.

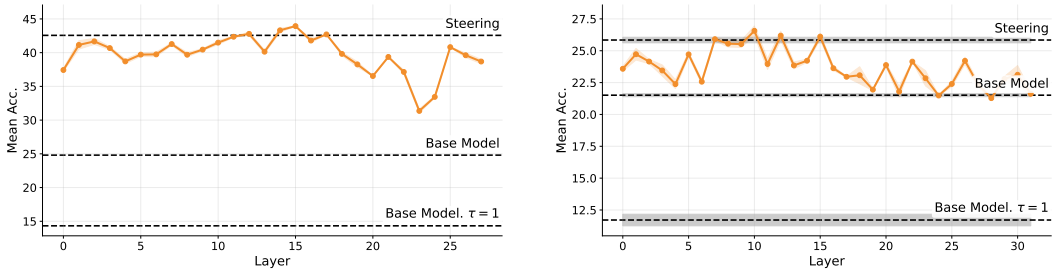
The results for LLaMa3.1-8B-It in Figure 6 mirror those for Qwen2.5-Math-7B in Section 2: mid-layer vectors perform best yet none reaches all-layer steering. Differently from Qwen2.5-Math-7B, the final-layer vector yields only marginal gains.

D Per-Layer Steering with Greedy Decoding

We evaluated the same single-layer steering vectors from Section 2 with temperature $\tau = 0$ and found that many match the performance of full-steering in this setup, see Figure 7. That shows that

[1] `https://github.com/huggingface/Math-Verify`

single-layer steering vectors modify the right mechanisms but lack the capacity to lower the entropy enough.



Qwen2.5-Math-7B LLaMa3.1-8B-It

Figure 7: **Single-Layer Steering with $\tau = 0$.**

We re-evaluate the single-layer steering vectors from Section 2 with greedy decoding ($\tau = 0$). As shown in Figure 7, many of these single-layer vectors match the performance of full steering, which we did not observe when sampling with temperature $\tau = 1$. This suggests they target the correct mechanisms but lack the capacity to sufficiently reduce generation entropy.

E Ineffective Layers in Qwen2.5-Math-7B

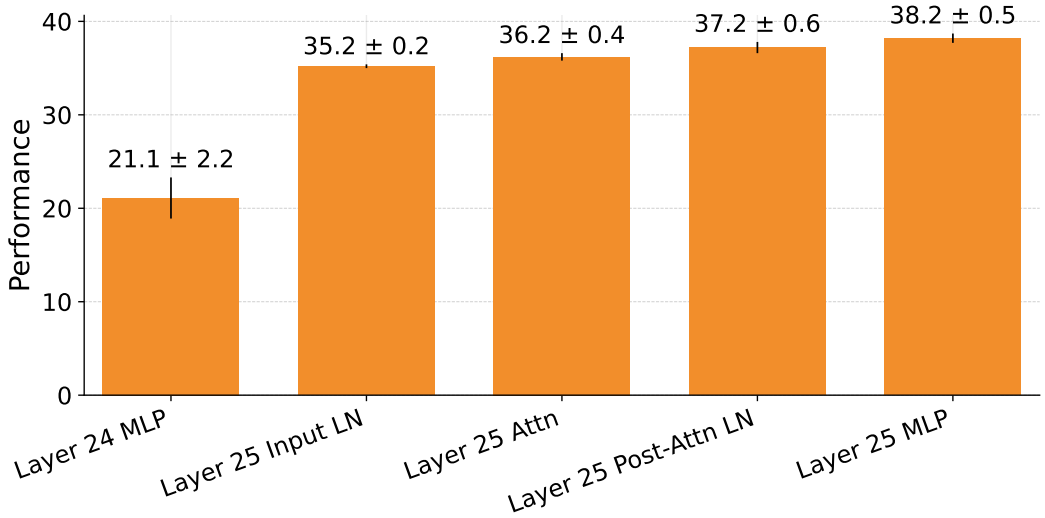


Figure 8: **Component-output steering.**

As noted in Section 2, single-layer steering on layers 23 and 24 underperforms their neighbors. To pinpoint where this loss arises, we trained vectors inserted immediately after each subcomponent between the layer-24 MLP and the layer-25 MLP. Figure 8 shows that placing s_{24} *after* the input LayerNorm of layer 25 closes the gap with s_{25}. Thus the input LayerNorm is the problematic step – passing through it limits the effect of the steering vector.

F Last Layer. Logit Lens

Table 1: **Last Layer – logit-lens.** Cosine similarities and dot-product scores between the last-layer steering vector (trained in isolation) and the unembedding vectors of the top-10 tokens for `Qwen2.5-Math-7B` and `LLaMa3.1-8B-It`.

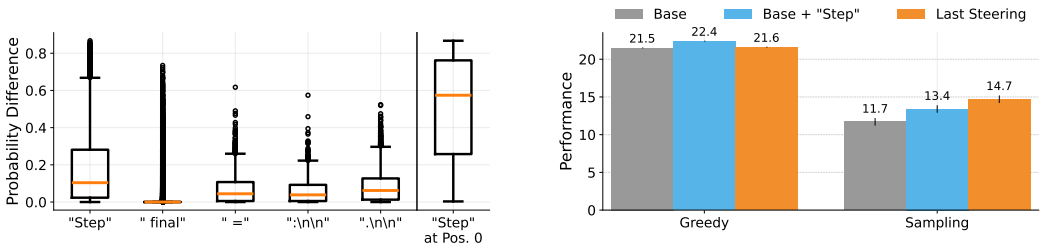
Qwen2.5-Math-7B

	`To`	`]`	`To`	`So`	`_to`	`\`	`}`	`For`	`.To`	`-to`
Cos. Sim.	0.37	0.16	0.16	0.15	0.14	0.14	0.13	0.13	0.12	0.12
Dot Prod.	42.5	19.12	18.62	19.12	16.88	19.75	15.69	14.19	17.0	18.62

LLaMa3.1-8B-It

	`final`	`Step`	`format`	`Final`	`final`	`final`	`Steps`	`Final`	`_final`	`solution`
Cos. Sim.	0.12	0.11	0.09	0.09	0.08	0.08	0.08	0.08	0.08	0.08
Dot Prod.	1.69	1.32	1.17	1.09	0.71	1.01	1.02	0.93	0.83	0.95

G Last Layer. LLaMa3.1-8B-It



(a) Distribution of the change in token probability (ΔP) produced by the single last-layer steering vector on Qwen-2.5-Math-7B. The box-plots summarise 256 DeepScaleR completions and display (i) the five tokens with the highest maximum ΔP and (ii) a separate distribution for "*To*" when it appears at the first generated position ("*To*" at Pos. 0).

(b) Prepending "*To*" to every prompt (Base + "*To*") raises performance by 10–11 absolute points under both greedy and sampling decoding by the base model – more than half of the gain achieved by explicit last-layer steering.

Figure 9: **Last Layer Analysis.** The left panel shows how last-layer steering concentrates its impact on starting with token "*To*"; the right panel confirms that this single-token boost translates into a substantial portion of the observed performance improvement.

We analyze the last-layer steering vector for LLaMa3.1-8B-It using the procedure in Section 3. Table 1 (see Section F) reports the `logit-lens` scores. Two observations stand out: (i) the vector is only weakly aligned with any single token – the largest cosine similarity is 0.12 – and (ii) the highest-scoring tokens are variations of "*final*" and "*Step*". Much of the vector's effect is concentrated on "*Step*" at the first generated position (Figure 9a).

Prepending "*Step*" to each prompt improves the pefromance of the *base* model under both *Sampling* and *Greedy* decoding. Interestingly, in the *Greedy* setting this prefix even outperforms last-layer steering, plausibly because a last-layer steering vector cannot condition its influence on position and thus perturbs subsequent steps.

H Value Steering Adds a Linear Term to MHA

The following derivation holds when we ignore the pre-attention LayerNorm (LN). While this is a strong assumption – that LN does not alter the steering vector's trajectory – the experiment in Section 4 shows that a post-attention steering vector attains the same performance as a pre-attention one, indicating that the pre-attention vector indeed does not act through attention.

Claim. Let $U \in \mathbb{R}^{T \times d_{\text{model}}}$ and define the (row-wise) attention

$$A(U) = \text{Softmax}\left(\frac{UW_i^Q (UW_i^K)^\top}{\sqrt{d_k}}\right) \in \mathbb{R}^{T \times T}, \quad A(U)\mathbf{1} = \mathbf{1}.$$

For head i,

$$H_i(U) = A(U)\, UW_i^V.$$

Let a steering vector $s \in \mathbb{R}^{d_{\text{model}}}$ be added to the *values* of head i for every token, and set $S = \mathbf{1}s^\top \in \mathbb{R}^{T \times d_{\text{model}}}$. Then

$$\begin{aligned} H_i^{(+s)}(U) &= A(U)\,(U + S)W_i^V \\ &= A(U)UW_i^V + A(U)SW_i^V \\ &= H_i(U) + SW_i^V \qquad (\text{since } A(U)\mathbf{1} = \mathbf{1}). \end{aligned}$$

Writing $W^O = \begin{bmatrix} W_1^O \\ \cdots \\ W_h^O \end{bmatrix}$ by heads, the multi-head output satisfies

$$\boxed{\text{MHA}(U + S) = \text{MHA}(U) + S\, W_i^V W_i^O}$$

and is independent of the attention pattern.

I Penultimate-Layer Steering Vector in `LLaMa3.1-8B-It`

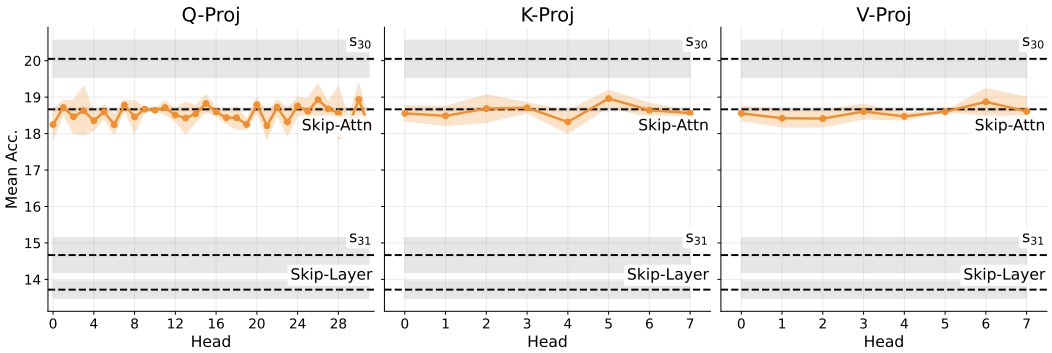


Figure 10: **Penultimate-layer steering in LLaMa3.1-8B-It.** Mean accuracy when the penultimate-layer vector s_{30} is injected into a single Q (left), K (center), or V (right) projection of the final block. Steering any single projection stays near *Skip-Attn* and below s_{30}.

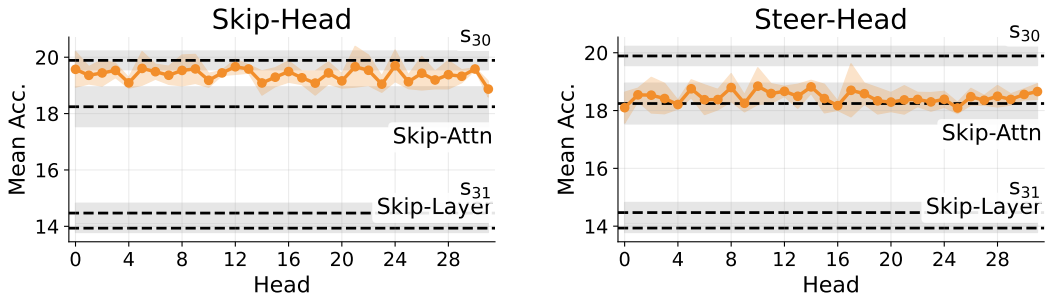


Figure 11: **Penultimate-layer steering in LLaMa3.1-8B-It.** Mean accuracy when applying s_{30} at the final block by patching whole heads: *Skip-Head* (left, steer all except head i) and *Steer-Head* (right, steer only head i). No single head closes the gap between *Skip-Attn* and s_{30}, indicating a cooperative multi-head effect.

Projection-level patching (**Steer–Q/K/V**) did not reveal the source of the gain (Figure 10). We therefore patched entire heads using two setups: **Steer–Head** ($H_i(U) \mapsto H_i(U + s_{l-1})$) and **Skip–Head** (leave $H_i(U)$ unchanged while steering all other heads). In Figure 11, two baselines mirror the Qwen result: *Skip-Layer* performs close to s_{31}, indicating a direct unembedding effect, and *Skip-Attn* retains about 70% of the s_{30} gain, suggesting much of the impact bypasses attention. No single head closes the remaining gap between s_{30} and *Skip–Attn*, pointing to a cooperative multi-head mechanism and the importance of attention layer for s_{30}'s performance; resolving this is left for future work.

However, training the steering vector in the post-attention residual stream yields performance indistinguishable from s_{30} (mean accuracy 19.9 ± 0.1), suggesting either that the vector effectively bypasses attention or that comparable performance can be achieved via the MLP alone.

J Pair Single Raw

Table 2: **Pairwise composition of steering vectors.** Mean accuracy (%) across six benchmarks when applying two independently trained steering vectors at once: s_i at layer i and s_j at layer j.

Qwen2.5-Math-7B

$i \backslash j$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0	34.6	35.6	36.0	37.1	36.2	36.7	36.4	38.2	38.4	39.1	38.7	39.9	39.9	40.3	41.6	41.7	41.9	40.0	40.8	39.4	38.7	39.4	38.4	37.5	37.2	39.5	39.7	37.9
1	—	35.6	34.9	35.1	36.0	36.1	35.2	36.7	37.4	37.8	38.3	39.6	38.5	39.3	40.4	40.8	41.0	39.2	39.5	39.4	38.9	39.0	38.6	37.2	37.4	39.9	39.0	37.1
2	—	—	35.2	36.3	36.2	36.2	36.0	37.6	37.6	38.0	38.4	39.1	39.5	40.3	41.0	41.7	41.6	39.3	40.3	39.7	39.0	39.1	38.9	36.6	36.9	39.8	39.8	38.0
3	—	—	—	36.3	32.7	36.0	35.6	38.0	37.8	37.8	39.0	40.2	39.6	40.1	40.9	41.4	41.5	39.4	40.2	40.1	38.5	39.8	39.2	37.5	37.7	40.5	39.6	38.1
4	—	—	—	—	35.4	36.4	35.4	37.4	37.0	38.1	38.0	39.6	39.2	40.2	41.3	41.6	42.3	39.7	40.2	40.0	38.6	38.9	39.1	37.6	36.6	40.6	39.5	38.0
5	—	—	—	—	—	37.0	35.8	37.2	37.4	38.4	38.5	39.6	40.1	40.0	41.1	41.4	41.5	39.6	41.1	40.8	39.8	39.5	39.3	38.3	37.4	40.1	39.7	37.9
6	—	—	—	—	—	—	36.3	36.7	36.4	37.7	37.8	39.4	38.8	39.4	41.0	41.4	41.5	39.8	40.3	39.7	39.0	39.6	38.9	37.5	36.9	40.5	39.9	37.8
7	—	—	—	—	—	—	—	37.1	36.0	37.1	38.1	39.0	39.5	39.8	41.7	41.6	41.9	40.3	40.2	41.1	40.1	40.2	39.4	38.3	37.7	41.1	40.3	38.5
8	—	—	—	—	—	—	—	—	36.9	35.9	37.3	38.8	38.2	39.4	40.8	40.3	41.5	39.9	40.0	40.7	40.0	40.4	39.0	38.0	38.4	40.1	41.0	38.3
9	—	—	—	—	—	—	—	—	—	37.4	37.4	38.5	39.3	39.5	39.7	40.7	40.9	39.1	39.2	40.2	40.3	40.4	40.0	38.8	38.2	40.8	40.6	39.5
10	—	—	—	—	—	—	—	—	—	—	37.2	37.1	38.6	39.6	40.3	40.6	40.8	40.2	41.1	40.2	40.5	40.9	39.6	37.7	37.6	41.0	41.2	39.6
11	—	—	—	—	—	—	—	—	—	—	—	37.4	38.0	38.9	39.3	40.6	41.6	39.3	40.2	40.7	40.2	40.8	39.8	38.3	37.5	40.9	41.4	39.9
12	—	—	—	—	—	—	—	—	—	—	—	—	37.2	37.3	39.4	40.5	41.1	39.4	40.4	40.6	40.1	40.8	40.2	39.1	37.9	41.9	41.1	39.8
13	—	—	—	—	—	—	—	—	—	—	—	—	—	38.6	39.1	40.1	41.2	39.7	40.4	40.9	40.2	40.7	39.8	38.3	38.1	41.3	40.0	40.5
14	—	—	—	—	—	—	—	—	—	—	—	—	—	—	39.9	39.3	40.4	39.5	40.4	41.6	41.6	41.2	40.2	39.8	39.1	43.1	42.2	41.6
15	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	39.2	39.8	38.5	39.5	40.7	41.0	41.0	39.9	39.6	39.1	42.5	40.0	42.2
16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	40.0	35.5	37.6	40.7	40.7	40.8	40.2	39.5	38.8	42.9	39.9	41.9
17	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	36.6	32.7	39.5	39.8	39.7	39.4	37.5	37.3	41.2	38.6	40.1
18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	37.5	36.9	37.9	38.0	38.0	34.8	36.9	39.8	34.7	40.1
19	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	38.2	31.1	34.6	35.6	26.1	35.4	39.1	25.5	39.5
20	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	36.3	33.4	31.6	18.3	33.9	38.1	18.3	37.9
21	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	37.4	32.6	34.3	35.3	36.8	24.0	38.3
22	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	36.5	13.8	24.5	27.0	17.4	32.2
23	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	23.2	15.5	35.6	12.1	33.7
24	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	24.6	34.2	12.9	34.2
25	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	38.8	20.7	35.3
26	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	37.5	36.4
27	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	30.6

LLaMa3.1-8B-It

$i \backslash j$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	21.1	17.3	17.4	17.7	18.1	19.2	19.1	19.5	21.2	22.2	20.5	22.6	22.5	20.6	21.9	21.1	22.0	20.1	19.1	22.1	19.7	20.1	22.0	20.3	20.6	19.5	19.8	20.2	19.8	20.0	20.5	20.6
1	—	21.9	17.1	17.9	17.5	17.9	18.4	19.9	20.4	22.7	20.5	22.7	22.4	20.9	21.9	20.7	21.0	20.2	20.7	21.2	20.9	19.3	21.8	20.9	21.1	19.6	20.2	21.2	20.1	19.7	20.7	20.9
2	—	—	22.8	18.3	18.6	18.7	19.3	19.2	20.6	22.8	20.5	23.0	22.9	20.7	21.8	21.1	21.7	20.9	20.6	22.0	21.4	21.2	22.5	21.4	22.2	20.6	21.2	21.7	21.1	20.7	22.0	21.6
3	—	—	—	22.6	18.5	19.4	19.5	19.4	20.8	23.8	21.6	23.5	23.2	20.8	22.8	21.7	22.6	20.8	20.8	22.6	21.4	21.4	22.5	22.0	21.8	19.6	21.1	21.4	21.0	20.1	22.4	21.6
4	—	—	—	—	21.7	18.1	19.0	19.4	19.6	22.9	20.9	22.2	22.6	20.6	22.4	20.8	21.0	20.1	20.1	20.8	21.3	20.2	21.6	20.9	21.6	12.8	18.2	20.6	20.8	11.3	21.0	21.5
5	—	—	—	—	—	22.3	18.8	19.0	20.7	22.1	20.7	22.3	22.9	20.5	21.8	20.5	21.6	20.8	21.5	22.4	21.4	21.2	22.7	21.5	22.0	19.3	20.7	21.1	21.0	20.7	21.9	21.3
6	—	—	—	—	—	—	21.8	18.5	19.9	21.5	19.4	21.9	22.1	20.6	20.7	20.4	21.6	20.9	21.4	20.8	21.7	20.9	22.3	21.5	21.7	19.4	20.6	20.8	20.8	20.7	21.5	20.7
7	—	—	—	—	—	—	—	22.4	18.9	19.4	18.6	21.0	21.6	20.0	20.9	20.4	21.6	20.4	20.6	20.9	21.0	20.3	22.4	20.8	21.6	14.5	19.1	21.0	20.1	18.3	21.5	20.7
8	—	—	—	—	—	—	—	—	23.0	20.9	18.6	21.1	22.4	20.9	23.0	21.6	23.3	22.4	22.7	23.2	23.2	23.7	24.2	23.6	23.6	21.9	22.4	23.5	24.0	23.8	23.8	23.1
9	—	—	—	—	—	—	—	—	—	24.7	18.7	20.4	23.6	21.4	22.9	22.7	24.4	23.0	22.7	23.7	24.0	23.7	24.9	24.0	24.7	23.3	23.4	23.7	24.3	23.7	24.4	23.8
10	—	—	—	—	—	—	—	—	—	—	23.4	17.6	20.3	19.3	20.9	20.5	23.1	21.7	21.8	22.5	22.6	22.6	23.6	22.5	23.0	21.3	20.8	22.0	22.2	22.4	22.9	22.4
11	—	—	—	—	—	—	—	—	—	—	—	24.2	20.9	20.8	21.7	21.6	23.7	22.7	23.1	23.8	24.3	23.3	24.1	24.0	24.2	22.4	23.3	24.0	24.1	24.5	24.1	24.0
12	—	—	—	—	—	—	—	—	—	—	—	—	24.5	20.8	22.5	21.7	23.4	22.7	23.3	23.8	24.2	23.5	24.3	23.9	23.7	22.2	23.5	23.6	23.3	23.1	24.1	23.0
13	—	—	—	—	—	—	—	—	—	—	—	—	—	23.1	20.0	19.5	20.9	20.0	20.6	20.3	21.1	20.7	20.7	21.3	21.6	19.6	20.6	21.5	21.0	20.6	21.7	21.9
14	—	—	—	—	—	—	—	—	—	—	—	—	—	—	24.2	19.7	22.0	21.3	21.9	22.3	23.4	22.9	24.3	23.4	23.6	21.5	23.0	23.5	23.3	22.2	23.5	23.4
15	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	23.5	21.1	21.3	22.1	21.0	23.0	22.3	23.0	22.6	23.1	21.1	22.5	22.7	23.0	22.0	22.5	23.1
16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	22.9	20.9	21.2	21.3	21.9	20.9	22.0	22.0	21.8	20.7	21.0	22.2	22.4	20.2	22.0	22.0
17	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	21.9	21.0	21.2	21.3	20.2	22.4	20.9	20.7	20.0	20.7	21.6	20.9	20.9	20.6	21.0
18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	21.5	20.2	20.7	20.7	21.9	20.4	20.5	19.4	20.6	20.4	19.5	20.4	19.9	20.6
19	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	22.8	21.0	21.0	21.0	21.6	20.8	19.7	21.1	21.2	20.5	20.4	21.3	22.2
20	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	21.9	20.5	21.3	20.7	20.5	19.6	20.9	21.9	20.6	20.4	20.9	20.8
21	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	21.1	21.7	19.7	19.5	19.4	20.4	20.8	19.1	19.6	20.0	20.5
22	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	22.3	21.9	21.8	20.3	21.3	22.4	21.2	21.1	21.2	21.9
23	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	21.1	20.0	19.3	19.6	20.9	19.1	16.6	20.1	20.1
24	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	21.4	19.0	19.0	20.4	18.8	17.2	20.2	20.6
25	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	21.9	13.7	18.6	19.2	16.6	20.0	21.4
26	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	21.7	18.8	19.6	16.2	20.0	20.7
27	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	21.6	19.5	15.3	20.2	20.0
28	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	20.9	13.2	19.2	20.8
29	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	21.1	12.3	20.3
30	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	20.0	18.5
31	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	14.8