HESITATION HAS A GEOMETRY: ENTROPY-TRAINED HYPERBOLIC PROBES FOR SPARSE ACTIVATION STEERING

Anonymous authors
Paper under double-blind review

ABSTRACT

When a large language model solves a mathematical problem, its reasoning is largely hierarchical, and the solution often branches at a few tokens where the next-token entropy is high. Such tree-like structure embeds in hyperbolic space with far lower distortion than in Euclidean space. Activation steering, however, usually edits the hidden states of a pretrained model by adding one fixed Euclidean vector at every token, even though most tokens of a solution are already determined by the context. We propose **H**yperbolic **E**ntropy **St**eering (HEST), which embeds the hidden states in the Poincaré ball with a lightweight probe whose only label is the model's own next-token entropy. Where this entropy exceeds a threshold, HEST moves the embedded state along the geodesic of steepest descent of a readout of the probe and maps the change back to the hidden state. For the Busemann readout of a learned ideal point, we prove that a step of fixed length lowers it by the same amount at every state. On three instruction-tuned models from the Qwen2.5-Math and Llama-3.1 families, HEST with the Busemann readout improves greedy accuracy on MATH-500 and GSM8K in five of six settings, by up to 1.8 points, whereas a contrastive steering vector added at every token lowers accuracy. With a Euclidean probe trained in the same way, this gain disappears on Qwen2.5-Math-1.5B-Instruct. The gains are largest on problems where the model hesitates often, and accuracy on the remaining problems is almost unchanged.

1 INTRODUCTION

Large language models solve mathematical problems through multi-step reasoning, and such reasoning, like language itself, is largely hierarchical. Hierarchical relations between concepts are reflected in the geometry of their representations (Park et al., 2025). In mathematical reasoning, a chain of thought also tends to branch at a small number of high-entropy tokens, which can determine the subsequent solution path (Wang et al., 2026). Hyperbolic space suits such tree-like structure, since its volume grows exponentially with the radius, and trees embed in it with far lower distortion than in a Euclidean space of the same dimension (Nickel & Kiela, 2017; Sala et al., 2018). This property has been used to model hierarchies in neural networks, Transformers and language models (Ganea et al., 2018; Yang et al., 2024b; 2026; He et al., 2026; Patil et al., 2026a;b), and it also makes hyperbolic space a natural candidate for representing the hidden states of Euclidean language models.

Activation steering changes the behavior of a pretrained language model at inference time without updating its weights. It shifts the hidden activations along a direction extracted from contrastive examples (Turner et al., 2023; Li et al., 2023; Zou et al., 2023; Rimsky et al., 2024). In activation-addition methods, this direction is a Euclidean vector that is typically the same at every token position (Figure 1a). Contrastive Activation Addition (CAA; Rimsky et al., 2024), for example, applies the same steering vector throughout generation. In mathematical reasoning, such a vector perturbs every token of a solution, although most tokens are largely determined by the preceding context and the solution branches at only a few high-entropy tokens. Existing steering methods for mathematical reasoning mostly act at the step boundaries of long reasoning models (Chen et al., 2025; Li et al., 2026b) and classify the steps by keywords such as "Wait", which instruction-tuned models rarely produce. We instead steer instruction-tuned language models only at high-entropy tokens, and we

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.

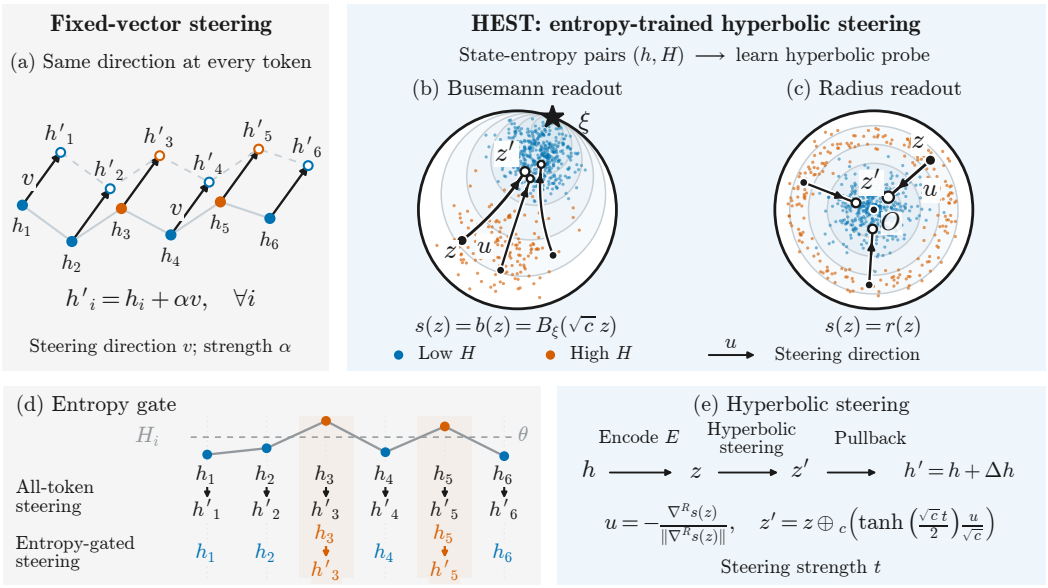


Figure 1: **HEST steers only the tokens at which the model hesitates, along a geodesic in the space of a hyperbolic probe.** (a) Fixed-vector steering adds the same vector αv at every token. (b, c) HEST maps hidden states into the Poincaré ball with a probe trained on next-token entropy. The readout s is either the Busemann function of the ideal point ξ or the distance r to the origin O, and the arrows show the steering direction u. (d) HEST steers only tokens whose entropy H_i exceeds the threshold θ. (e) At such a token, h is mapped to z, moved by hyperbolic distance t along the geodesic in the direction u, and mapped back. In (e), $\|\cdot\|$ is the Euclidean norm. The point clouds and step lengths in (b, c) are schematic, and Appendix O shows real embedded states.

take the direction from a hyperbolic embedding of their hidden states that a lightweight probe learns from the same entropy.

We start from an observation about the model's own uncertainty. The entropy of the next-token distribution is low at most positions and exceeds a threshold at only a small fraction of them, which we call *hesitation points*. The same entropy can also serve as a training signal, and we train a probe that maps hidden states into the Poincaré ball and orders them by entropy. Without external labels, it learns a geometry in which confident states lie closer than hesitant ones to an ideal point on the boundary of the ball, or to its origin. The Busemann function of this ideal point and the distance to the origin serve as readouts, and the negative Riemannian gradient of a readout gives a steering direction that depends on the state. We build Hyperbolic Entropy Steering (HEST) on this observation (Figure 1). During greedy decoding, HEST embeds the hidden state at every hesitation point and moves the embedding along a geodesic that lowers the readout. It then maps the change back to the hidden state, decodes the next token from the edited state and edits no other position. We show that a step of hyperbolic length t lowers the Busemann readout by exactly $\sqrt{c}\,t$, where $-c$ is the curvature, and the radius readout by exactly t whenever the step does not pass the origin. Hence, the step length has the same meaning at every state (Proposition 1).

We evaluate HEST on Qwen2.5-Math-1.5B-Instruct, our development model, Qwen2.5-Math-7B-Instruct and Llama-3.1-8B-Instruct, using MATH-500, GSM8K and, for the development model, OlympiadBench. With the Busemann readout, HEST improves greedy accuracy in five of the six model and dataset pairs on MATH-500 and GSM8K, by up to 1.8 points, while steering a median of 0.8 to 3.0% of solution tokens. A difference-of-means vector built from the same states and added at every token, as in CAA, lowers accuracy in all six pairs. With a Euclidean probe trained in the same way and edits of the same size, the Busemann readout also lowers accuracy for the development model. The gains are largest on the few problems on which the greedy solution hesitates often. Most other greedy solutions are already correct, and HEST rarely changes their outcome. Our contributions are:

- We propose HEST, an inference-time steering method in which the model's own next-token entropy decides where to steer and also trains the hyperbolic probe that gives the direction. HEST requires no correctness labels, contrastive prompts, language model training or additional sampled solutions.
- We give a geometric account of the method. Both readouts have a direct interpretation, one geodesic step lowers a readout by an exact amount, and in the probe space the Busemann direction depends on the state, whereas its Euclidean counterpart does not.
- We evaluate HEST on three models against greedy decoding and CAA, ablate where to steer, in which direction and in which geometry, and analyze which problems benefit.

2 Related Work

Activation steering. Most activation steering methods add a fixed vector to the hidden states, at every generated token or at fixed positions of the prompt. The vector is obtained by contrasting prompts or examples on behaviors such as truthfulness, refusal and sentiment (Turner et al., 2023; Li et al., 2023; Zou et al., 2023; Rimsky et al., 2024). ReFT and HyperSteer instead learn the intervention, as low-rank edits at fixed prompt positions and as vectors that a hypernetwork generates from a steering prompt and the input (Wu et al., 2024; Sun et al., 2025). Several recent methods let the edit depend on the state. AcT transports activations with coordinate-wise affine maps from optimal transport (Rodriguez et al., 2025), and ODESteer and FLAS move them along a vector field (Zhao et al., 2026; Jin et al., 2026). Curveball Steering works in the feature space of a kernel PCA (Raval et al., 2026), and Angular and Spherical Steering rotate activations (Vu & Nguyen, 2026; You et al., 2026). K-Steering, the closest to our mechanism, follows the gradient of a nonlinear classifier trained with attribute labels (Oozeer et al., 2025). CLAS scales a direction at every layer by a coefficient linear in the activation, and it uses GSM8K to elicit step-by-step solutions (Hsu et al., 2026). Other methods restrict where the edit is applied. MAT-Steer learns a gate that weights each steering vector at each prompt token (Nguyen et al., 2025), and Spherical Steering rotates only when a confidence score for the undesired direction exceeds a threshold (You et al., 2026). FASB backtracks when a probe trained on labeled data detects a deviation (Cheng et al., 2026). Helix lowers the largest logit where a score of the next-token entropy and a Mahalanobis distance of the hidden state indicates divergence (Atkinson, 2026). HEST combines a state-dependent direction with a sparse gate, and both come from the model's own entropy rather than from labels, as the comparison in Appendix A details.

Entropy and forking tokens. Token-level entropy marks the positions where a reasoning chain often branches. In reinforcement learning for reasoning, a minority of high-entropy tokens determines the reasoning path, and restricting policy updates to these tokens matches or exceeds updating all tokens (Wang et al., 2026). Hidden states also reflect uncertainty about which path the generation will take (Zur et al., 2025). HARP treats tokens whose entropy exceeds a threshold as points of hesitation, applies dropout to the input embeddings and mixes the logits of a second forward pass (Storaï & Hwang, 2025). Entropy-gated branching instead selects among several continuations at such tokens with a process reward model (Li et al., 2026a). Some steering methods for mathematical reasoning target long reasoning models. SEAL splits the reasoning into steps at double line breaks, classifies the steps by keywords and steers the representations at every step boundary (Chen et al., 2025). FlowSteer keeps this protocol but replaces the linear edit with flow matching (Li et al., 2026b). Sun et al. (2026) study the reasoning trajectories of three Llama-3.1-8B variants and also steer at step boundaries. Their steering uses a predictor trained on correctness labels, or ideal trajectories from correct solutions. Like HARP and entropy-gated branching, HEST selects tokens by their entropy, but it edits their hidden states and uses neither keywords nor correctness labels.

Latent-space methods for test-time scaling. Another line of work spends additional inference compute in latent space. Fractional Reasoning scales a latent steering vector to control the depth of reasoning, and it reports its gains with majority voting or best-of-N selection (Liu et al., 2025). LatentSeek optimizes latent representations for each problem with a policy gradient at test time (Li et al., 2025). SPREAD jointly optimizes steering vectors for several sampled solutions to diversify them, by Riemannian optimization over a product of spheres (Khanh et al., 2026). These methods use several candidates or optimize per problem, whereas HEST edits a single greedy decode, so we

do not use them as baselines. Random embedding injection appends untrained random embeddings to the input to diversify reasoning, and it is complementary to HEST (Kim et al., 2026).

Hyperbolic representation. Ganea et al. (2018) define hyperbolic neural network layers through Möbius gyrovector operations in the Poincaré ball. Ghadimi Atigh et al. (2021) use Busemann functions to classify against ideal prototypes, and Chen et al. (2026) build hyperbolic network layers from them. HypFormer builds fully hyperbolic Transformers (Yang et al., 2024b). HypLoRA fine-tunes large language models with low-rank adapters in hyperbolic space (Yang et al., 2026), and HELM builds hyperbolic large language models, including a variant whose experts have different curvatures (He et al., 2026). Patil et al. (2026a) survey hyperbolic language models. HyperGuide projects hidden states into the Poincaré ball and fine-tunes the model to use the resulting signal in multi-step reasoning (Liu et al., 2026). Our probe follows the ranking and triplet objective of HyperGuide, which trains its projection on distances to a solution in a search tree rather than on the model's own entropy. We use our readouts to steer a Euclidean model without fine-tuning it.

3 METHOD

Setup and notation. Let M be a decoder-only language model, decoded greedily, with residual stream $h \in \mathbb{R}^d$ at layer ℓ. After the model reads the token at position i, it has a layer-ℓ state h_i and a distribution p_i over the next token, with entropy

$$H_i \;=\; -\sum_{v \in \mathcal{V}} p_i(v) \log p_i(v). \tag{1}$$

We call position i a *hesitation point* when $H_i > \theta$. The entropy plays two roles in HEST: the probe learns to order states by it (Section 3.1), and it decides where to steer (Section 3.4).

3.1 A HYPERBOLIC PROBE TRAINED ON ENTROPY

We use the Poincaré ball $\mathbb{D}_c^n = \{z \in \mathbb{R}^n : c\|z\|^2 < 1\}$ of curvature $-c$, with metric $g_z = (\lambda_z^c)^2 g^E$ and conformal factor $\lambda_z^c = 2/(1 - c\|z\|^2)$ (Ganea et al., 2018). The probe $E : \mathbb{R}^d \to \mathbb{D}_c^n$ is

$$E(h) = \exp_0^c\left(\frac{\phi\big((h-\mu)/\sigma\big)}{\omega}\right), \qquad \exp_0^c(v) = \tanh\big(\sqrt{c}\,\|v\|\big)\,\frac{v}{\sqrt{c}\,\|v\|}, \tag{2}$$

where μ and σ are the coordinate-wise mean and standard deviation of the training states, and ϕ is a two-layer perceptron with hidden width 1024 and output dimension $n = 128$. The scalar ω is a running root mean square of $\|\phi(\cdot)\|$ and fixes the scale of the embedding.

Readouts. A readout is a scalar function s on the ball, low at confident states and high at hesitant ones. The first readout is built from the Busemann function of a learned ideal point on the boundary of the ball (Ghadimi Atigh et al., 2021). We write this point as $\xi/\sqrt{c}$ with $\|\xi\| = 1$ and define

$$b(z) = B_\xi\big(\sqrt{c}\,z\big), \qquad B_\xi(y) = \log\frac{\|\xi - y\|^2}{1 - \|y\|^2}. \tag{3}$$

Here B_ξ is the Busemann function of the unit ball, so b is $\sqrt{c}$ times that of $\mathbb{D}_c^n$. Its level sets are horospheres, which are spheres tangent to the boundary at the ideal point, and b decreases without bound along every geodesic that ends at $\xi/\sqrt{c}$. The second readout is the distance to the origin,

$$r(z) = d_c(0, z) = \frac{2}{\sqrt{c}}\,\operatorname{artanh}\big(\sqrt{c}\,\|z\|\big), \tag{4}$$

whose level sets are spheres centered at the origin. Both readouts are interpretable, as confident states tend to lie on smaller horospheres than hesitant ones, or nearer to the origin (Figure 1b,c).

Training objective. Write $z_i = E(h_i)$. We follow Liu et al. (2026), who train a hyperbolic probe with a pairwise margin ranking loss (Herbrich et al., 2000) and a triplet loss (Weinberger & Saul, 2009; Schroff et al., 2015) in the Poincaré ball. The probe minimizes

$$\mathcal{L} = \mathop{\mathbb{E}}_{(i,j):\, H_i < H_j - \delta}\Big[\gamma + s(z_i) - s(z_j)\Big]_+ + \mathop{\mathbb{E}}_{(a,p,q)}\Big[\gamma + d_c(z_a, z_p) - d_c(z_a, z_q)\Big]_+, \tag{5}$$

with $[x]_+ = \max(0, x)$, $\delta = 0.3$ nats and $\gamma = 0.1$. The first term orders states by entropy, and the second keeps nearby states of one solution close: the positive p lies within five tokens of the anchor a, and the negative q comes from another problem. The curvature and the ideal point are learned jointly with ϕ. The data are the states and entropies of greedy solutions to 100 training problems, without any correctness label. On held-out problems, the Busemann readout orders 85 to 90% of the entropy pairs correctly in all settings, and the radius readout 82 to 89% (Appendix D).

3.2 Steering along a Geodesic

Let h be the state at a hesitation point and $z = E(h)$. Because the metric is conformal, the Riemannian gradient $\nabla^{\mathcal{R}} s(z) = (\lambda_z^c)^{-2} \nabla s(z)$ has the direction of the Euclidean gradient. We set

$$u = -\frac{\nabla^{\mathcal{R}} s(z)}{\|\nabla^{\mathcal{R}} s(z)\|}, \qquad z' = z \oplus_c \left(\tanh\Big(\frac{\sqrt{c}\, t}{2}\Big) \frac{u}{\sqrt{c}} \right), \tag{6}$$

where $\|\cdot\|$ is the Euclidean norm and $\oplus_c$ is Möbius addition (Ganea et al., 2018). The point z' equals $\exp_z^c(t\, v)$ with $v = u/\lambda_z^c$, the unit tangent vector in the direction u under the hyperbolic metric.

Proposition 1. *Let z' be given by Eq. 6 with $t > 0$ and $\nabla s(z) \neq 0$. Then (i) $d_c(z, z') = t$; (ii) for the radius readout, if $z \neq 0$ and $t \leq r(z)$, then $r(z') = r(z) - t$; (iii) for the Busemann readout, $b(z') = b(z) - \sqrt{c}\, t$.*

Part (i) follows from the left-cancellation law of Möbius addition. Parts (ii) and (iii) use two standard facts: distance and Busemann functions have unit gradient in the hyperbolic metric, and their descent curves are geodesics (Bridson & Haefliger, 2013). The factor $\sqrt{c}$ comes from the rescaling $y = \sqrt{c}\, z$, and Appendix B gives the proof. The step length t thus has the same meaning at every state. The condition $t \leq r(z)$ holds at every hesitation point of all greedy test solutions (Appendix B.3).

Role of the curvature. In the limit $c \to 0$, evaluated in Section 5, the probe becomes Euclidean, the geodesic step becomes a straight step $z' = z + (t/2)\, u$, and the readouts become the norm and the linear function $-\langle \xi, z \rangle$ up to scale (Appendix N). For the radius readout, the step points to the origin in both geometries, and the two differ only in how far a step of length t moves. The Busemann readout is where the geometries differ. In the Poincaré ball, its descent curves are geodesics that converge on the ideal point, so the direction u depends on the state. The linear function, in contrast, has parallel hyperplanes as level sets and the same descent direction ξ everywhere (Appendix B.5). After the mapping back to the hidden state, any readout gives a state-dependent direction in $\mathbb{R}^d$ through the Jacobian of the probe (Appendix L), so the geometries differ in the probe space.

3.3 Mapping the Step back to the Hidden State

We look for an edit Δh with $E(h + \Delta h) \approx z'$ and solve this underdetermined problem by damped Gauss–Newton iterations from $\Delta h = 0$,

$$\Delta h \;\leftarrow\; \Delta h + J^\top \big(J J^\top + \eta I\big)^{-1} \big(z' - E(h + \Delta h)\big), \qquad J = \frac{\partial E}{\partial h}\Big|_{h+\Delta h}, \tag{7}$$

with $\eta = 10^{-3} \operatorname{tr}(J J^\top)/n$. We run at most four iterations and clip $\|\Delta h\|$ to at most $\|h\|$. Each iteration solves an $n \times n$ system, and J has a closed form. The edited states realize the distance t and the drop $\sqrt{c}\, t$ of the Busemann readout in Proposition 1 to within 10^{-4} on average (Appendix C).

3.4 Decoding with the Entropy Gate

HEST runs inside greedy decoding, where the model produces p_i after reading the token y_i. If $H_i > \theta$, we remove the key and value entries of position i from the cache. We then recompute the forward pass of y_i with $h_i + \Delta h_i$ in place of h_i at layer ℓ and choose the next token from the new distribution. The edit thus acts on the state that produced the uncertain distribution, and one position later the model is usually confident again (Appendix E). The edited keys and values stay in the cache, so later tokens attend to the edited state. No other position is edited (Algorithm 1). A steered decode that reaches the limit of 3072 new tokens falls back to the greedy answer.

3.5 Choice of Hyperparameters

The three hyperparameters of HEST are the layer ℓ, the threshold θ and the step length t, and we take the layer from prior work. For the Qwen models, we use $\ell = 19$, the layer that SEAL uses for DeepSeek-R1-Distill-Qwen-1.5B and 7B, which are fine-tuned from Qwen2.5-Math (Chen et al., 2025). For Llama, we use $\ell = 13$, the layer that CAA uses for Llama 2 7B Chat and ODESteer uses for Llama-3.1-8B (Rimsky et al., 2024; Zhao et al., 2026), and Appendix D explains the layer indexing. The threshold is $\theta = 1$ for the development model, Qwen2.5-Math-1.5B-Instruct. Since $\exp(H_i)$ is the effective number of candidate tokens, $H_i > 1$ means more than $e \approx 2.7$ effective candidates. Because entropy scales differ across models, we choose θ for each other model so that HEST steers about as many tokens per solution as on the development model. We estimate this number before steering from the teacher-forced entropies of the greedy solutions, which needs no labels (Appendix G). The step length is chosen on MATH-500 and GSM8K for the development model (Table 1): $t = 0.5$ for the Busemann readout and $t = 0.25$ for the radius readout. We keep these values for the other two models, so no label of their test sets enters their hyperparameters.

4 Empirical Results

4.1 Setup

Models and benchmarks. We use Qwen2.5-Math-1.5B-Instruct as the development model, together with Qwen2.5-Math-7B-Instruct (Yang et al., 2024a) and Llama-3.1-8B-Instruct (Grattafiori et al., 2024). We evaluate on MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024), the 1,319 test problems of GSM8K (Cobbe et al., 2021) and, for the development model, OlympiadBench (He et al., 2024). From OlympiadBench, we take the 675 English, text-only mathematics problems with open-ended answers, of which a fixed set of 100 trains the probe and the other 575 are used for evaluation. OlympiadBench is harder, and the model hesitates on it about twice as densely as on MATH-500, so we use it to test HEST under dense hesitation (Section 4.3). For MATH-500 and GSM8K, the probe is trained on 100 problems from the training split of the same dataset. All methods decode greedily from the same prompt with a limit of 3072 new tokens. All outputs are graded by one protocol that extracts the final answer and compares it with the reference (Appendix F).

Baselines. We compare with greedy decoding and with CAA (Rimsky et al., 2024). The CAA direction is the difference of means $v = \mathbb{E}[h \mid H \le \theta] - \mathbb{E}[h \mid H > \theta]$. It uses the same probe-training states, layer and threshold as HEST. It is added at every generated token with multiplier 1, as in most experiments of the original method. CAA is thus the fixed-direction, all-token counterpart of HEST, built from the same data. We report it without the fallback of Section 3.4, and with the fallback it still lowers accuracy (Appendix I). We do not include SEAL and FlowSteer, which rely on keyword-labeled steps of long reasoning models. We also do not include the method of Sun et al. (2026), which needs correctness labels, or methods that modify the input or use several samples. State-dependent methods such as K-Steering, ODESteer and CLAS learn from labels or contrastive examples (Appendix A), so Section 5 instead trains the probe of HEST with $c \to 0$.

Table 1: Accuracy (%) of Qwen2.5-Math-1.5B-Instruct and change from greedy decoding (points) at four step lengths t. Greedy decoding: 74.20 on MATH-500, 84.84 on GSM8K and 41.39 on OlympiadBench. All other models use the step lengths $t^\star$. Bold: best accuracy per benchmark.

	HEST, Busemann ($t^\star = 0.5$)			HEST, radius ($t^\star = 0.25$)		
t	MATH-500	GSM8K	OlympiadBench	MATH-500	GSM8K	OlympiadBench
0.1	74.80 (+0.60)	84.53 (−0.30)	41.39 (0.00)	74.20 (0.00)	84.53 (−0.30)	41.04 (−0.35)
0.25	75.00 (+0.80)	84.84 (0.00)	41.39 (0.00)	75.20 (+1.00)	84.53 (−0.30)	40.87 (−0.52)
0.5	**76.00** (+1.80)	**85.37** (+0.53)	40.00 (−1.39)	74.40 (+0.20)	84.38 (−0.45)	42.26 (+0.87)
1	75.40 (+1.20)	85.29 (+0.45)	41.22 (−0.17)	73.40 (−0.80)	84.15 (−0.68)	**42.61** (+1.22)

4.2 Main Results

For the development model (Table 1), HEST with the Busemann readout improves MATH-500 accuracy at every step length, by up to 1.80 points at $t = 0.5$, and $t = 0.5$ is also the best step length on GSM8K. Averaged over the four step lengths, it still gains 1.10 points on MATH-500 and 0.17 points on GSM8K. The radius readout depends more on the step length on MATH-500, where it gains 1.00 point at $t = 0.25$ and loses 0.80 points at $t = 1$, and it does not improve GSM8K. The Busemann readout does not improve OlympiadBench at any step length (Section 4.3).

Table 2 applies the same configuration to the other models, with the threshold calibrated as in Section 3.5 and the step lengths unchanged. With the Busemann readout, HEST improves greedy accuracy in five of the six model and dataset pairs, by 0.23 to 1.80 points, including three of the four pairs whose hyperparameters involve no test labels. The exception is Llama-3.1-8B-Instruct on GSM8K, with 18 problems fixed and 22 broken. Over all six pairs, HEST fixes 88 problems and breaks 63 (Appendix H gives all counts and paired sign tests). The radius readout gives smaller changes, between -0.30 and $+1.00$ points. CAA lowers accuracy in all six settings, by 0.38 to 12.60 points, although it uses the same data. It changes 97 to 100% of the solutions, fixing 205 problems and breaking 369 (Appendix I). HEST, in contrast, steers a median of 0.8 to 3.0% of the tokens of a solution (Appendix K), and each steered token costs one extra single-token forward pass of the model, besides the probe, its Jacobian and at most four $n \times n$ solves.

Table 2: Accuracy (%) and change from greedy decoding (points). θ: gate threshold. Steered: median percentage of tokens steered per solution (Busemann). Bold: best accuracy per row.

				HEST			
Model	Dataset	θ	Greedy	Busemann	radius	CAA	Steered (%)
Qwen2.5-Math-1.5B	MATH-500	1.0	74.20	**76.00** (+1.80)	75.20 (+1.00)	70.00 (−4.20)	1.1
	GSM8K	1.0	84.84	**85.37** (+0.53)	84.53 (−0.30)	84.23 (−0.61)	1.6
Llama-3.1-8B	MATH-500	2.0	50.80	**52.00** (+1.20)	51.00 (+0.20)	38.20 (−12.60)	0.8
	GSM8K	1.4	**84.61**	84.31 (−0.30)	84.53 (−0.08)	80.06 (−4.55)	3.0
Qwen2.5-Math-7B	MATH-500	1.0	84.00	**84.80** (+0.80)	84.20 (+0.20)	82.60 (−1.40)	1.0
	GSM8K	0.8	95.38	**95.60** (+0.23)	95.45 (+0.08)	95.00 (−0.38)	2.0

4.3 Analysis of the Main Results

Hesitation groups. To see which problems benefit, Figure 2 divides them by the number of hesitation points in their greedy solution, which is fixed before any intervention, into groups with 0 to 10, 11 to 24 and at least 25 hesitation points. Let g be the greedy accuracy of a group, f the fraction of its greedy-wrong problems that HEST fixes, and b the fraction of its greedy-right problems that HEST breaks. The change in accuracy, in points, is then $\Delta = 100\,[(1-g)\,f - g\,b]$. The figure shows g and Δ for every group, and Appendix J lists f and b.

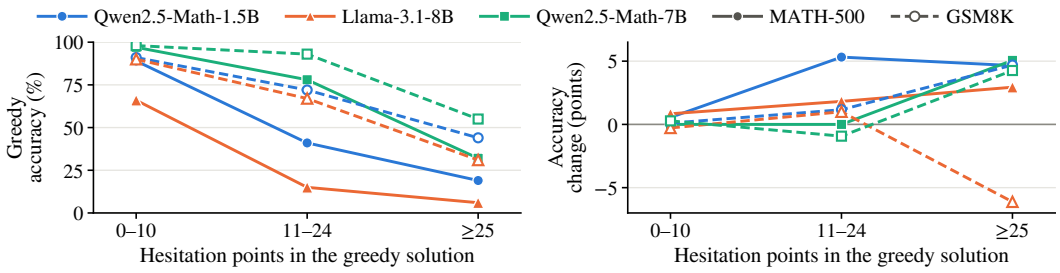
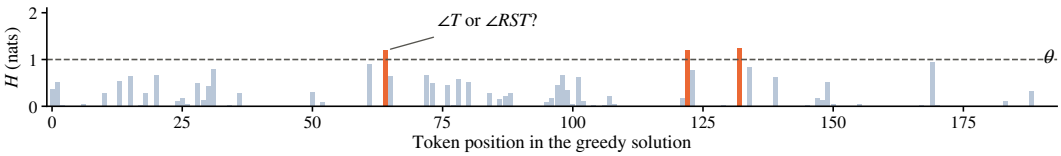


Figure 2: Accuracy changes little on most problems, and HEST gains most where the greedy solution hesitates often. Problems are grouped by the number of hesitation points in the greedy solution. Left: greedy accuracy. Right: change in accuracy with HEST, Busemann readout, $t = 0.5$.

First, most problems hesitate little and are already solved. In all six settings, the 0–10 group contains 71 to 81% of the problems, and in five of the six settings greedy decoding solves 89 to 98% of them. HEST breaks almost nothing in this group ($b \leq 0.03$), and accuracy changes by -0.28 to $+0.84$ points. Second, the gain per problem tends to grow with hesitation. In the group with at least 25 hesitation points, HEST gains 2.94 to 5.06 points in five of the six settings, but this group is small.

Dense hesitation. The two largest losses, on OlympiadBench and for Llama-3.1-8B-Instruct on GSM8K, occur in groups where greedy decoding solves only about a third of the problems. On OlympiadBench, 34% of the problems have at least 25 hesitation points, against 9% on MATH-500, and HEST steers 20 tokens per solution on average, against 9 (Appendix K). More edits give more chances to change a correct derivation, and in the middle group the Busemann readout breaks 22% of the solved problems. The radius readout breaks 8% and gains up to 1.22 points at larger steps (Table 1), so the setting chosen on MATH-500 and GSM8K does not carry over to dense hesitation.

An example. Figure 3 follows the MATH-500 problem with the shortest greedy solution among those that HEST fixes for the development model. The first of its three hesitation points is where the model states which angle is the right angle. Greedy decoding writes $\angle T$, contrary to the diagram, and concludes $\sin T = 1$. After one geodesic step at this token, the model writes $\angle RST$ and eventually reaches $\sqrt{21}/5$. In most other fixed problems, the first changed token is a choice of wording.



Problem (MATH-500, geometry, level 3). In $\triangle RST$, shown, $\sin R = \frac{2}{5}$. What is $\sin T$? *[Diagram: right angle at S; RT = 5.]*

Common prefix. … Given: … $\triangle RST$ is a right triangle with $\angle$? first hesitation point, $H = 1.19 > \theta$

Greedy decoding	HEST: one geodesic step at this token
$\angle T = 90^\circ$ … "since $\angle T = 90^\circ$, we know that $\sin 90^\circ = 1$" … $\boxed{1}$ ✗	$\angle RST = 90^\circ$ … "$\angle R$ and $\angle T$ are complementary" … $\sin T = \cos R = \boxed{\frac{\sqrt{21}}{5}}$ ✓

Figure 3: One steered token changes the course of the solution. Top: next-token entropy along the greedy solution, in which 3 of 189 tokens exceed $\theta = 1$. Bottom: the two continuations from the first of them. HEST steers 12 of the 583 tokens of its own solution. Appendix P gives both solutions.

5 Ablation Study

On Qwen2.5-Math-1.5B-Instruct, we ablate where to steer, in which direction and in which geometry. Each variant of HEST changes a single component and keeps layer 19, the fallback and, unless stated otherwise, the Busemann readout (Table 3). Appendix M also varies the threshold and the layer, which were fixed before the ablation, and neither improves on $\theta = 1$ and layer 19.

Where to steer. Two variants apply a step of the same size along the probe direction at other positions. *Every token* edits all generated positions, and *step boundaries* edits the tokens that contain a double line break, where the reasoning steps of SEAL begin. Neither variant improves accuracy on MATH-500 (Table 3a), and steering step boundaries lowers it by 0.40 points. It edits about as many tokens as HEST, but only 8.2% of the hesitation points fall at the start of a step. CAA adds a vector about five times larger than the HEST step at every token and lowers accuracy by 4.20 points. A small step at every token thus stays close to neutral, and only edits at hesitation points yield a gain.

Table 3: Ablations on Qwen2.5-Math-1.5B-Instruct with $\theta = 1$ and layer 19, in accuracy (%) and change from greedy decoding (points). (a) Where to steer, on MATH-500 with the Busemann readout, $t = 0.5$. Steered: median fraction of steered tokens. Step: median $\|\Delta h\|/\|h\|$. CAA has no fallback. (b) Which direction, with the gate and step size of HEST. (c) Which geometry, where the Euclidean probe has $c \to 0$ and the same median edit size. Bold marks the best of each comparison.

(a) Positions	Steered	Step	Accuracy
Hesitation points (HEST)	1.1%	0.06	**76.00** (+1.80)
Every token	100%	0.07	74.00 (−0.20)
Step boundaries	1.4%	0.07	73.80 (−0.40)
CAA, every token	100%	0.33	70.00 (−4.20)

(b) Direction	MATH-500	GSM8K
Greedy	74.20	84.84
Probe (HEST)	**76.00** (+1.80)	**85.37** (+0.53)
Fixed	**76.00** (+1.80)	84.91 (+0.08)
Reverse	74.60 (+0.40)	84.99 (+0.15)

	MATH-500		GSM8K	
(c) Readout	Hyperbolic (HEST)	Euclidean	Hyperbolic (HEST)	Euclidean
Busemann	**76.00** (+1.80)	73.40 (−0.80)	**85.37** (+0.53)	84.69 (−0.15)
radius	**75.20** (+1.00)	74.40 (+0.20)	**84.53** (−0.30)	**84.53** (−0.30)

Which direction. Two further variants keep the gate and the step size and change only the direction. *Fixed direction* uses the unit CAA direction, scaled to the median size of the HEST edit, and *reverse direction* steps away from the ideal point (Table 3b). The reverse direction gains only 0.40 and 0.15 points over greedy decoding, and the fixed direction matches HEST on MATH-500 but gains only 0.08 points on GSM8K, against 0.53. With the gate and the small step, these three directions break at most 4.4% of the changed solutions that were correct. The direction decides how many wrong solutions are fixed, and HEST fixes the largest fraction of the changed wrong solutions on both benchmarks, 15% on MATH-500 and 20% on GSM8K. A fixed direction is thus competitive under the gate and the step size of HEST, but not when CAA adds it at every token.

Which geometry. The Euclidean probe, the limit $c \to 0$ of the probe of HEST, omits the exponential map and is trained in the same way. It predicts entropy at least as well as the hyperbolic probe. Its step length matches the median edit size of HEST on training states (Appendix N). With the Busemann readout, the Euclidean probe lowers accuracy by 0.80 points on MATH-500 and by 0.15 points on GSM8K, whereas HEST gains 1.80 and 0.53 points (Table 3c) and improves MATH-500 at every step length of Table 1. Over the two benchmarks, only HEST is right on 48 problems, and only the Euclidean probe on 26. It fixes a smaller fraction of the changed wrong solutions, 11% against 15% on MATH-500 and 14% against 20% on GSM8K, and on MATH-500 it breaks 9% of the changed correct ones, against 4%. With the radius readout, whose step points to the origin in both geometries, the two probes differ by 0.80 points on MATH-500 and match on GSM8K. The curvature thus matters most for the Busemann readout, as Section 3.2 predicts.

6 Limitations and Conclusion

Limitations. The gains of HEST are bounded by how many problems can still be fixed, and near the ceiling of greedy decoding, as for Qwen2.5-Math-7B-Instruct on GSM8K, there is little room to improve. When solutions hesitate densely, as on many OlympiadBench problems, correct answers are fragile and HEST can break them. Other decoding methods, models with more than 8B parameters and tasks beyond mathematical reasoning remain to be tested.

Conclusion. HEST steers a language model with hyperbolic geometry, using the model's own next-token entropy as its only label. The entropy selects the few tokens at which the model hesitates, and a probe trained on the same signal gives the direction through the Riemannian gradient of its readout. Steering only these tokens improves greedy accuracy on mathematical reasoning in most settings, whereas a fixed vector added at every token lowers it. On the development model, the gain disappears with a Euclidean probe that ranks entropy at least as well, which suggests that it depends on the hyperbolic geometry.

AI USE STATEMENT

The research idea, the method and the plan of the experiments come from the authors. We did not use generative AI tools to develop the idea or the method. We also did not use them to derive or prove the mathematical results, or to generate, clean or reformat data. Translation and qualitative data analysis do not apply to this work.

We used generative AI tools as assistants to help us with the code for the method, the baselines and the evaluation. They also gave feedback on the experimental setup and helped us analyze the results. In addition, we used them to aid and polish writing for clarity and grammar. In all of these tasks, the authors decided what to run, what to report and how to interpret the results. We reviewed all AI-assisted work. We read and tested the code, and we checked every number and claim in the text against the outputs of our runs. The authors are responsible for the entire content of this paper, including the text and code written with AI assistance.

ETHICS STATEMENT

This work studies an inference-time intervention on the hidden states of language models for mathematical problem solving. The same mechanism could in principle be used to steer other behaviors. The only label of its probe is the model's own next-token entropy, which carries no task semantics. The method as described therefore does not target any specified behavior. All datasets are public benchmarks of mathematics problems and contain no personal data. We also report OlympiadBench, on which the selected configurations do not improve accuracy, together with an analysis of the reason.

REPRODUCIBILITY STATEMENT

Section 3 specifies the probe, the readouts, the geodesic step and the mapping back to the hidden state in closed form, and Appendix B proves Proposition 1. Appendix C gives the Jacobian of the probe, Appendix D the training details, Appendix E the decoding procedure, Appendix F the grading protocol and Appendix G the calibration of the threshold. The layer is taken from prior work, the threshold is set by a rule that uses no labels, and the step length is chosen on the development model only (Section 3.5). The code, the configuration of every run and the Busemann probes of the development model are available at `https://anonymous.4open.science/r/HEST-code` and in the supplementary material.

REFERENCES

Craig Atkinson. Tethered reasoning: Decoupling entropy from hallucination in quantized LLMs via manifold steering. *arXiv preprint arXiv:2602.17691*, 2026.

Martin R Bridson and André Haefliger. *Metric spaces of non-positive curvature*. Springer Science & Business Media, 2013.

Runjin Chen, Zhenyu Zhang, Junyuan Hong, Souvik Kundu, and Zhangyang Wang. SEAL: Steerable reasoning calibration of large language models for free. In *Second Conference on Language Modeling*, 2025. URL `https://openreview.net/forum?id=klPszYDIRT`.

Ziheng Chen, Bernhard Schölkopf, and Nicu Sebe. Hyperbolic Busemann neural networks. In *CVPR*, 2026.

Zifeng Cheng, Jinwei Gan, Zhiwei Jiang, Cong Wang, Yafeng Yin, Xiang Luo, Yuchen Fu, and Qing Gu. Steering when necessary: Flexible steering large language models with backtracking. *Advances in Neural Information Processing Systems*, 38:8390–8413, 2026.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in neural information processing systems*, 31, 2018.

Mina Ghadimi Atigh, Martin Keller-Ressel, and Pascal Mettes. Hyperbolic Busemann learning with ideal prototypes. *Advances in neural information processing systems*, 34:103–115, 2021.

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.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen 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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Neil He, Rishabh Anand, Hiren Madhu, Ali Maatouk, Smita Krishnaswamy, Leandros Tassiulas, Menglin Yang, and Rex Ying. HELM: Hyperbolic large language models via mixture-of-curvature experts. *Advances in Neural Information Processing Systems*, 38:142604–142635, 2026.

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.

Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Large margin rank boundaries for ordinal regression. In Alexander J. Smola, Peter Bartlett, Bernhard Schölkopf, and Dale Schuurmans (eds.), *Advances in Large Margin Classifiers*, pp. 115–132. MIT Press, 2000. doi: 10.7551/mitpress/1113.003.0010.

Brandon Hsu, Daniel Beaglehole, Adityanarayanan Radhakrishnan, and Mikhail Belkin. Contextual linear activation steering of language models. *arXiv preprint arXiv:2604.24693*, 2026.

Zehao Jin, Ruixuan Deng, Junran Wang, Xinjie Shen, and Chao Zhang. Beyond steering vector: Flow-based activation steering for inference-time intervention. In *Mechanistic Interpretability Workshop at ICML 2026*, 2026. URL https://openreview.net/forum?id=t7M7vjIJbe.

Ly Tran Ho Khanh, Dongxuan Zhu, Man-Chung Yue, and Viet Anh Nguyen. Test-time diverse reasoning by Riemannian activation steering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 31429–31437, 2026.

Heejun Kim, Seungpil Lee, Jewon Yeom, Jaewon Sok, Seonghyeon Park, Jeongjae Park, Taesup Kim, and Sundong Kim. From noise to diversity: Random embedding injection in LLM reasoning. *arXiv preprint arXiv:2605.11936*, 2026.

Hynek Kydlíček. Math-Verify: Math verification library, 2025. URL https://github.com/huggingface/math-verify.

Hengli Li, Chenxi Li, Tong Wu, Xuekai Zhu, Yuxuan Wang, Zhaoxin Yu, Eric Hanchen Jiang, Song-Chun Zhu, Zixia Jia, Ying Nian Wu, et al. Seek in the dark: Reasoning via test-time instance-level policy gradient in latent space. *arXiv preprint arXiv:2505.13308*, 2025.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in neural information processing systems*, 36:41451–41530, 2023.

Xianzhi Li, Ethan Callanan, Abdellah Ghassel, and Xiaodan Zhu. Entropy-gated branching for efficient test-time reasoning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5054–5069, 2026a.

Yawei Li, Benjamin Bergner, Yinghan Zhao, Vihang Prakash Patil, Bei Chen, and Cheng Wang. Steering large reasoning models towards concise reasoning via flow matching. *Transactions on Machine Learning Research*, 2026b. ISSN 2835-8856. URL https://openreview.net/forum?id=qwcJMdGerK.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Sheng Liu, Tianlang Chen, Pan Lu, Haotian Ye, Yizheng Chen, Lei Xing, and James Zou. Fractional reasoning via latent steering vectors improves inference time compute. *arXiv preprint arXiv:2506.15882*, 2025.

Yuyu Liu, Haotian Xu, Yanan He, Sarang Rajendra Patil, Mengjia Xu, and Tengfei Ma. HyperGuide: Hyperbolic guidance for efficient multi-step reasoning in large language models. *arXiv preprint arXiv:2605.24140*, 2026.

Duy Nguyen, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Multi-attribute steering of language models via targeted intervention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20619–20634, 2025.

Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017.

Narmeen Fatimah Oozeer, Luke Marks, Fazl Barez, and Amir Abdullah. Beyond linear steering: Unified multi-attribute control for language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 23513–23557, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.1278. URL https://aclanthology.org/2025.findings-emnlp.1278/.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models. In *International Conference on Learning Representations*, volume 2025, pp. 76441–76463, 2025.

Sarang Patil, Zeyong Zhang, Yiran Huang, Tengfei Ma, and Mengjia Xu. Hyperbolic large language models. *SIAM Review (To appear)*, 2026a.

Sarang Rajendra Patil, Ashish Parmanand Pandey, Ioannis Koutis, and Mengjia Xu. Hierarchical Mamba meets hyperbolic geometry: A new paradigm for structured language embeddings. *Transactions on Machine Learning Research*, 2026b. ISSN 2835-8856. URL https://openreview.net/forum?id=a3g13FKzct.

Shivam Raval, Hae Jin Song, Linlin Wu, Abir Harrasse, Jeff M Phillips, Fazl Barez, and Amirali Abdullah. Curveball steering: The right direction to steer isn't always linear. *arXiv preprint arXiv:2603.09313*, 2026.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, 2024.

Pau Rodriguez, Arno Blaas, Michal Klein, Luca Zappella, Nicholas Apostoloff, Xavier Suau, et al. Controlling language and diffusion models by transporting activations. In *International Conference on Learning Representations*, volume 2025, pp. 89812–89855, 2025.

Frederic Sala, Chris De Sa, Albert Gu, and Christopher Ré. Representation tradeoffs for hyperbolic embeddings. In *International conference on machine learning*, pp. 4460–4469. PMLR, 2018.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

Romain Storaï and Seung-won Hwang. Harp: Hesitation-aware reframing in transformer inference pass. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 12305–12319, 2025.

Jiuding Sun, Sidharth Baskaran, Zhengxuan Wu, Michael Sklar, Christopher Potts, and Atticus Geiger. HyperSteer: Activation steering at scale with hypernetworks. *arXiv preprint arXiv:2506.03292*, 2025.

Lihao Sun, Hang Dong, Bo Qiao, Qingwei Lin, Dongmei Zhang, and Saravan Rajmohan. LLM reasoning as trajectories: Step-specific representation geometry and correctness signals. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 26872–26887, 2026.

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.

Minh Hieu Vu and Tan Nguyen. Angular steering: Behavior control via rotation in activation space. *Advances in Neural Information Processing Systems*, 38:121653–121690, 2026.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. *Advances in Neural Information Processing Systems*, 38:115452–115486, 2026.

Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research*, 10(2), 2009.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. ReFT: Representation finetuning for language models. *Advances in Neural Information Processing Systems*, 37:63908–63962, 2024.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2.5-Math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024a.

Menglin Yang, Harshit Verma, Delvin Ce Zhang, Jiahong Liu, Irwin King, and Rex Ying. Hypformer: Exploring efficient transformer fully in hyperbolic space. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3770–3781, 2024b.

Menglin Yang, Aosong Feng, Bo Xiong, Jiahong Liu, Irwin King, Rex Ying, et al. Hyperbolic fine-tuning for large language models. *Advances in Neural Information Processing Systems*, 38: 29613–29648, 2026.

Zejia You, Chunyuan Deng, and Hanjie Chen. Spherical steering: Geometry-aware activation rotation for language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=9PPeJsVLdB.

Hongjue Zhao, Haosen Sun, Jiangtao Kong, Xiaochang Li, Qineng Wang, Liwei Jiang, Qi Zhu, Tarek F. Abdelzaher, Yejin Choi, Manling Li, and Huajie Shao. ODESteer: A unified ODE-based steering framework for LLM alignment. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=CFewUmgIIL.

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.

Amir Zur, Atticus Geiger, Ekdeep Singh Lubana, and Eric Bigelow. Are language models aware of the road not taken? token-level uncertainty and hidden state dynamics. *arXiv preprint arXiv:2511.04527*, 2025.

Appendix

A Comparison with Related Steering Methods **14**

B Hyperbolic Geometry and Proof of Proposition 1 **16**

C Closed-Form Jacobian of the Probe **18**

D Probe Training Details **18**

E Decoding Procedure **19**

F Grading Protocol **20**

G Calibration of the Gate Threshold **20**

H Full Results **21**

I CAA Baseline **22**

J Hesitation Groups and Hesitation Density **22**

K How Many Tokens HEST Steers **23**

L Diversity of the Steering Directions **24**

M Ablation Details **25**

N Euclidean Probes **25**

O Probe States on Real Data **27**

P The Example of Section 4.3 in Full **28**

A Comparison with Related Steering Methods

Table 4 compares HEST with the steering methods discussed in Section 2 along the design choices studied in this paper. Most methods edit every token or fixed positions of the prompt, and MAT-Steer weights its edits of the prompt tokens with a learned gate. The few methods that restrict the positions during generation use a confidence score computed from a contrastive direction, a probe trained on labels, step boundaries, or a score that combines the next-token entropy with a distance to a reference distribution. Helix uses the last of these scores to lower the largest logit and does not edit the hidden state. HEST restricts the positions with the entropy alone, and the same entropy trains the probe that gives the direction. Most methods also derive their direction from contrastive or labeled data. Among the methods that report accuracy on mathematical benchmarks, SEAL and FlowSteer target long reasoning models, and the method of Sun et al. (2026) needs correctness labels. CLAS uses GSM8K to elicit step-by-step solutions, ReFT trains its interventions on the task, Helix evaluates a quantized model at high sampling temperatures, and Fractional Reasoning and SPREAD report accuracy over several samples.

Table 4: Steering methods compared along the design choices of HEST. *Positions*: the tokens at which the edit is applied. *Math*: whether accuracy on mathematical benchmarks is reported, and in which setting. Each entry was checked against the original paper.

Method	Direction	Positions	Training signal	Geometry	Evaluation	Math
CAA (Rimsky et al., 2024)	fixed, difference of means	all tokens after the prompt	contrastive pairs	Euclidean	behavioral questions, open-ended generation, MMLU, TruthfulQA	–
ITI (Li et al., 2023)	fixed, top attention heads	every generated token	labeled truthful and untruthful answers	Euclidean	TruthfulQA, NQ, TriviaQA, MMLU	–
CLAS (Hsu et al., 2026)	per-layer direction, coefficient linear in the activation	all tokens	probe direction, prompt and completion pairs	Euclidean	11 tasks including GSM8K	GSM8K, to elicit step-by-step solutions
ReFT (Wu et al., 2024)	learned low-rank edit of a subspace	fixed prefix and suffix positions of the prompt	task supervision, frozen model	Euclidean	commonsense and arithmetic reasoning, instruction following, GLUE	GSM8K and three others, after task fine-tuning
HyperSteer (Sun et al., 2025)	generated by a hypernetwork from the steering prompt and the input	all tokens	AxBench steering prompts, language-modeling loss	Euclidean	AxBench concepts	–
AcT (Rodriguez et al., 2025)	coordinate-wise affine maps from optimal transport	all tokens	source and target sentence sets	Euclidean	toxicity, concept induction, TruthfulQA, text-to-image models	–
ODESteer (Zhao et al., 2026)	classifier gradient, integrated as an ODE	all generated tokens	positive and negative examples	Euclidean	UltraFeedback, TruthfulQA, RealToxicityPrompts	–
FLAS (Jin et al., 2026)	concept-conditioned velocity field	all tokens	positive examples, language-modeling loss	Euclidean	AxBench concepts	–
Curveball (Raval et al., 2026)	fixed direction in kernel PCA space	all generated tokens	contrastive pairs	kernel feature space	behavioral choices, tone traits	–
K-Steering (Oozeer et al., 2025)	classifier gradient	all tokens	attribute labels	Euclidean	tone and debate style, MMLU as a capability check	–
Angular (Vu & Nguyen, 2026)	rotation in a fixed plane	all tokens; adaptive variant where aligned with the feature	contrastive prompt sets	sphere	refusal, emotion	tinyGSM8K, as a capability check
Spherical (You et al., 2026)	geodesic rotation towards a fixed direction	each token, strength set by a confidence gate	contrastive answer pairs	sphere	TruthfulQA and five multiple-choice benchmarks	–
MAT-Steer (Nguyen et al., 2025)	fixed per attribute	learned soft gate per prompt token	labeled positive and negative samples	Euclidean, norm preserving	TruthfulQA, Toxigen, BBQ, HelpSteer	–
FASB (Cheng et al., 2026)	fixed per head	probe-triggered, with backtracking	labeled probe	Euclidean	TruthfulQA, six multiple-choice tasks	–
Helix (Atkinson, 2026)	penalty on the largest logit, no edit of the hidden state	tokens with low confidence from entropy and Mahalanobis distance	activations of low-temperature generations on factual prompts	Euclidean, Mahalanobis	GSM8K, HumanEval, MMLU, creative ideation	GSM8K, 4-bit model, temperatures 0.5 to 3.0
SEAL (Chen et al., 2025)	fixed	step boundaries	steps classified by keywords	Euclidean	MATH-500, GSM8K, LiveCodeBench	long reasoning models
FlowSteer (Li et al., 2026b)	flow matching	step boundaries	as SEAL	Euclidean	MATH-500, GSM8K, AIME, AMC, LiveCodeBench	long reasoning models
Sun et al. (2026)	fixed, or towards a centroid	step boundaries or before the answer, gated	correctness labels	Euclidean	GSM8K, MATH-500, MMLU	Llama-3.1-8B-Instruct
Fractional Reasoning (Liu et al., 2025)	fixed, scaled, norm preserved	all tokens	contrastive instruction prompts	Euclidean	GSM8K, MATH-500, GPQA	with voting or best-of-N
SPREAD (Khanh et al., 2026)	optimized jointly for N samples	tokens after each synchronization anchor	none	Euclidean, fixed vector norms	AIME, MATH-500, OlympiadBench	Pass@N
HEST (ours)	Riemannian gradient of a probe readout	hesitation points	the model's own entropy	hyperbolic	MATH-500, GSM8K, OlympiadBench	three instruction-tuned models

State-dependent methods as baselines. We do not run the state-dependent methods of Table 4 as baselines. K-Steering, ODESteer, FLAS and CLAS learn their edits from attribute labels, contrastive examples or positive examples of a target behavior, and hesitation has no such data apart from the entropy split. Among them, only CLAS reports results on GSM8K, where it elicits step-by-step solutions from prompts that ask only for the answer. Retraining each of them on the entropy split would change the methods themselves. We therefore build the closest state-dependent baseline from our own pipeline. The Euclidean probe of Appendix N is trained on the same entropy and lowers a nonlinear readout of the hidden state, much as K-Steering and ODESteer follow the gradient of a nonlinear classifier, and it differs from HEST only in the curvature. CAA plays the same role for a fixed direction, since we build it from the same entropy split.

B Hyperbolic Geometry and Proof of Proposition 1

Throughout, $c > 0$ is the curvature magnitude, so that the sectional curvature is $-c$, and n is the dimension of the ball.

B.1 The Poincaré Ball

The Poincaré ball model is

$$\mathbb{D}_c^n = \{z \in \mathbb{R}^n : c\|z\|^2 < 1\}, \qquad g_z = (\lambda_z^c)^2\, g^E, \qquad \lambda_z^c = \frac{2}{1 - c\|z\|^2}, \tag{8}$$

where g^E is the Euclidean metric. The metric is a positive multiple of the Euclidean one at every point, so the model is conformal: angles in g and in g^E agree. The boundary sphere is at infinite hyperbolic distance from every interior point. In the gyrovector formulation of Ganea et al. (2018), the Möbius addition of $x, y \in \mathbb{D}_c^n$ is

$$x \oplus_c y = \frac{\big(1 + 2c\langle x, y\rangle + c\|y\|^2\big)\, x + \big(1 - c\|x\|^2\big)\, y}{1 + 2c\langle x, y\rangle + c^2\|x\|^2\|y\|^2}, \tag{9}$$

and the distance is

$$d_c(x, y) = \frac{2}{\sqrt{c}} \operatorname{artanh}\big(\sqrt{c}\,\|(-x) \oplus_c y\|\big). \tag{10}$$

The exponential map at x follows the geodesic that leaves x with initial velocity $v \in T_x\mathbb{D}_c^n$ for unit time. It is given by

$$\exp_x^c(v) = x \oplus_c \left(\tanh\Big(\frac{\sqrt{c}\,\lambda_x^c\|v\|}{2}\Big)\frac{v}{\sqrt{c}\,\|v\|}\right). \tag{11}$$

At the origin $\lambda_0^c = 2$, and Eq. 11 reduces to the map used in the probe, Eq. 2. For a smooth function f on the ball, the Riemannian gradient is $\nabla^{\mathcal{R}} f(z) = (\lambda_z^c)^{-2}\nabla f(z)$, and its length in the hyperbolic metric is $\|\nabla^{\mathcal{R}} f(z)\|_g = (\lambda_z^c)^{-1}\|\nabla f(z)\|$.

B.2 Proof of Part (i)

Let u have unit Euclidean norm and $w = \tanh(\sqrt{c}\,t/2)\, u/\sqrt{c}$, so that $z' = z \oplus_c w$ and $\sqrt{c}\,\|w\| = \tanh(\sqrt{c}\,t/2)$. Möbius addition satisfies the left-cancellation law $(-z) \oplus_c (z \oplus_c w) = w$ (Ganea et al., 2018). By Eq. 10,

$$d_c(z, z') = \frac{2}{\sqrt{c}} \operatorname{artanh}\big(\sqrt{c}\,\|w\|\big) = \frac{2}{\sqrt{c}} \operatorname{artanh}\Big(\tanh\big(\tfrac{\sqrt{c}\,t}{2}\big)\Big) = t. \tag{12}$$

Moreover, $z' = \exp_z^c(t\, v)$ with $v = u/\lambda_z^c$. This vector has unit length in the hyperbolic metric, since $\|v\|_g = \lambda_z^c\|v\| = 1$, and substituting $t\, v$ into Eq. 11 gives w. □

B.3 Proof of Part (ii): the Radius Readout

The gradient of $r(z) = \frac{2}{\sqrt{c}} \operatorname{artanh}(\sqrt{c}\|z\|)$ is $\nabla r(z) = \lambda_z^c\, z/\|z\|$ for $z \neq 0$, so $\|\nabla^{\mathcal{R}} r\|_g = 1$ and the steering direction is $u = -z/\|z\|$. Then w is a negative multiple of z. For collinear vectors $x = \rho_1\hat{e}$ and $y = \rho_2\hat{e}$ with $\|\hat{e}\| = 1$, Eq. 9 simplifies: the numerator is $(\rho_1 + \rho_2)(1 + c\rho_1\rho_2)\,\hat{e}$ and the

denominator is $(1 + c\rho_1\rho_2)^2$, so $x \oplus_c y = \frac{\rho_1+\rho_2}{1+c\rho_1\rho_2}\,\hat{e}$. Writing $\sqrt{c}\,\rho_1 = \tanh\alpha$ and $\sqrt{c}\,\rho_2 = \tanh\beta$ gives $\sqrt{c}\,\frac{\rho_1+\rho_2}{1+c\rho_1\rho_2} = \tanh(\alpha+\beta)$. With $\hat{e} = z/\|z\|$, $\alpha = \sqrt{c}\,r(z)/2$ and $\beta = -\sqrt{c}\,t/2$, the new point is $z' = \tanh\big(\sqrt{c}\,(r(z)-t)/2\big)\,\hat{e}/\sqrt{c}$. For $0 < t \le r(z)$, this gives $r(z') = r(z) - t$. □

The condition $t \le r(z)$ in our experiments. Part (ii) requires the step to be no longer than the radius readout of the state. We evaluate $r(z)$ at every hesitation point of the greedy test solutions, using the state that HEST edits when its gate fires. These states coincide with the edited states at the first hesitation point of a solution and approximate them afterwards. Over the 117,032 states of the seven radius probes, the smallest value of $r(z)$ for each probe lies between 1.13 and 1.55 for Qwen2.5-Math-1.5B-Instruct, whose largest step length is $t = 1$. It lies between 0.88 and 1.19 for the other two models, which use $t = 0.25$. None of these states therefore violates the condition, and every radius step taken at them ends before the origin.

B.4 Proof of Part (iii): the Busemann Readout

A Busemann function is a renormalized distance to an ideal point on the boundary at infinity. For a geodesic ray γ with unit speed that ends at this point, it is $B(y) = \lim_{s\to\infty}\big(d(y,\gamma(s)) - s\big)$ (Bridson & Haefliger, 2013, Chapter II.8). First take $c = 1$. The function $B_\xi(y) = \log\big(\|\xi - y\|^2/(1-\|y\|^2)\big)$ is the Busemann function of the unit Poincaré ball associated with the ideal point ξ, normalized so that $B_\xi(0) = 0$ (Ghadimi Atigh et al., 2021; Bridson & Haefliger, 2013). Its level sets are horospheres, and a short calculation shows that the level set $\{B_\xi = \log k\}$ is the Euclidean sphere with center $\xi/(1+k)$ and radius $k/(1+k)$, which is tangent to the boundary at ξ.

The descent behavior of B_ξ is easiest to see in the upper half-space model $\mathbb{H}^n = \{x \in \mathbb{R}^n : x_n > 0\}$ with metric $\|dx\|^2/x_n^2$. A Möbius transformation of $\mathbb{R}^n \cup \{\infty\}$ maps the unit ball isometrically onto $\mathbb{H}^n$ and can be chosen to send ξ to the point at infinity. Busemann functions are defined by distances alone, so they are preserved by isometries, and the Busemann function of the point at infinity is $B_\infty(x) = -\log x_n$ up to an additive constant, which follows directly from the definition of Busemann functions (Bridson & Haefliger, 2013, Chapter II.8). Its Euclidean gradient is $-e_n/x_n$, so its Riemannian gradient is $x_n^2(-e_n/x_n) = -x_n e_n$, which has length 1 in the metric. The direction of steepest descent is therefore $+e_n$ at every point, and the geodesic that leaves a point in this direction is the vertical line through it. Along this line, hyperbolic arc length is $ds = dx_n/x_n$, so moving a hyperbolic distance τ multiplies x_n by e^τ and lowers B_∞ by exactly τ. Transferring back to the ball, one step of hyperbolic length τ along the steepest-descent geodesic of B_ξ lowers B_ξ by exactly τ.

For general c, the dilation $z \mapsto \sqrt{c}\,z$ sends $\mathbb{D}_c^n$ onto the unit ball. Since $\lambda^1_{\sqrt{c}z} = \lambda^c_z$, the metric of the unit ball pulls back to c times the metric of $\mathbb{D}_c^n$. The dilation therefore multiplies all distances by $\sqrt{c}$ and maps geodesics to geodesics. For the readout $b(z) = B_\xi(\sqrt{c}\,z)$, it sends the steepest-descent geodesic of b at z to that of B_ξ at $\sqrt{c}\,z$. A step of length t in $\mathbb{D}_c^n$ becomes a step of length $\sqrt{c}\,t$ in the unit ball. By the previous paragraph, $b(z') = b(z) - \sqrt{c}\,t$. □

B.5 Remarks on the Direction

The metric does not change the initial direction. Because the metric is conformal, the Riemannian gradient is a positive multiple of the Euclidean gradient, and after normalization both give the same u. The geometry enters the method in two other places. The step follows a geodesic of $\mathbb{D}_c^n$ and covers a fixed hyperbolic distance (Proposition 1). The embedding itself is also shaped by the geometry, because the probe is trained with the hyperbolic distance in Eq. 5 and with a readout defined on the ball.

The Busemann direction depends on the state. The probe is defined in the coordinates of the ball, and the step is mapped back to the hidden state from these coordinates. In the ball, the steepest-descent geodesics of B_ξ are the diameter through ξ and circular arcs that meet the boundary orthogonally at ξ, so the direction u differs from state to state. This dependence is not an artifact of the coordinates. In the upper half-space model, the descent directions look parallel, but the unit field $V = x_n\partial_n$ is not parallel in the hyperbolic sense. For $n = 2$, the Christoffel symbols of the metric give $D_{\partial_1}V = -\partial_1 \neq 0$, where D is the Levi-Civita connection. Parallel transport along a horo-

sphere therefore does not preserve the direction of steepest descent. The Euclidean counterpart of b, a linear function, has a constant gradient, which is a parallel field.

C Closed-Form Jacobian of the Probe

The update of Eq. 7 needs $J = \partial E/\partial h$ at every iteration. Because E is a fixed composition, J has a closed form. Write the forward pass of Eq. 2 as

$$x = \frac{h-\mu}{\sigma}, \quad a = W_1 x + b_1, \quad m = a \odot \Phi(a), \quad q = \frac{W_2 m + b_2}{\omega}, \quad z = \exp_0^c(q), \tag{13}$$

where Φ is the standard normal distribution function, so that $m = \mathrm{GELU}(a)$ in its exact form. With Φ' the standard normal density, $\partial m_k/\partial a_k = \Phi(a_k) + a_k \Phi'(a_k)$ and

$$\frac{\partial q}{\partial h} = \omega^{-1}\, W_2 \,\mathrm{diag}\big(\Phi(a) + a \odot \Phi'(a)\big)\, W_1 \,\mathrm{diag}(1/\sigma). \tag{14}$$

Let $\nu = \|q\|$ and $f(\nu) = \tanh(\sqrt{c}\,\nu)/(\sqrt{c}\,\nu)$, so that $z = f(\nu)\, q$. Then

$$\frac{\partial z}{\partial q} = f(\nu)\, I_n + \frac{f'(\nu)}{\nu}\, q q^\top, \qquad f'(\nu) = \frac{\mathrm{sech}^2(\sqrt{c}\,\nu)}{\nu} - \frac{\tanh(\sqrt{c}\,\nu)}{\sqrt{c}\,\nu^2}, \tag{15}$$

and $J = (\partial z/\partial q)(\partial q/\partial h)$, where ω is fixed after training. The damping in Eq. 7 is relative to the scale of the problem, $\eta = 10^{-3}\,\mathrm{tr}(JJ^\top)/n$. It keeps the $n \times n$ solve stable when $JJ^\top$ is ill-conditioned, for example near the boundary of the ball, where the radial derivative $\mathrm{sech}^2(\sqrt{c}\,\nu)$ of the exponential map becomes much smaller than its tangential derivative $f(\nu)$.

Accuracy of the mapping back. We reran the MATH-500 decodes of the development model with the Busemann readout while logging every steered token, and the rerun reproduces all 500 solutions. Over the $4{,}426$ steered tokens in 483 solutions, the mean drop of the readout per solution lies between 0.4293 and 0.4295, against $\sqrt{c}\,t = 0.4294$ in Proposition 1. The mean hyperbolic distance between z and $E(h + \Delta h)$ lies between 0.5000 and 0.5001, against $t = 0.5$. The edited states thus realize the distance and the drop of the readout that Proposition 1 prescribes to within 10^{-4} on average in every solution. In the 56 solutions with a single steered token, the drop lies between 0.4293 and 0.4294 and the distance is 0.5000, so the same holds for individual edits. The Euclidean probes of Appendix N lower their readouts by the prescribed amounts equally closely.

D Probe Training Details

Layer indexing. We index layers by the output of a decoder block, starting from zero, so our layer ℓ is `hidden_states[`$\ell + 1$`]` in the Hugging Face convention. SEAL adds its vector to the input of decoder layer 20 of DeepSeek-R1-Distill-Qwen-1.5B and 7B, which is the output of our layer 19 (Chen et al., 2025). CAA edits the output of the block with index 13 in Llama 2 7B Chat (Rimsky et al., 2024). ODESteer reports layer 14 of Llama-3.1-8B, and its code, which counts layers from zero, uses index 13 (Zhao et al., 2026). The CAA and ODESteer settings both correspond to our layer 13.

Data. For each model and dataset, we decode 100 training problems greedily once, with the prompt used at evaluation. For every generated position, we record the layer-ℓ state and the entropy of the next-token distribution. The state at the last prompt position emits the first generated token, and we align states and entropies accordingly. The number of states ranges from $15{,}821$ to $87{,}813$ per dataset (Table 5).

Length limit of the training decodes. This step only collects training states, so its length limit plays a different role from the limit of 3072 new tokens used in all evaluations (Section 4.1). The limit is 1024 new tokens for Qwen2.5-Math-1.5B-Instruct on MATH and GSM8K, 3072 on OlympiadBench and 2048 for the other two models, which is at least twice the median length of the training solutions in every setting. It cuts off 9 of the 100 solutions for Qwen2.5-Math-1.5B-Instruct on MATH, 6 on OlympiadBench and 5 for Llama-3.1-8B-Instruct on MATH, and no solution in the

other settings. Because the model is causal, a cut leaves every recorded state and entropy unchanged, and a longer limit would only add positions at the ends of these solutions. These ends are nearly deterministic, since over the last 200 tokens of the cut solutions the mean entropy is at most 0.19 nats and at most 6.5% of the tokens exceed 1 nat. A longer limit would therefore add few hesitation points and mostly confident states.

Split, pairs and triplets. Twenty of the 100 problems are held out for checkpoint selection. The split is by problem, because states within one solution are strongly correlated. Ranking pairs (i, j) are sampled subject to $H_i < H_j - \delta$ with $\delta = 0.3$ nats. Triplet anchors and positives come from the same solution within five tokens of each other, and negatives come from a different problem. Each optimization step uses 4096 pairs and 4096 triplets drawn from pools of $400{,}000$ pairs and $200{,}000$ triplets.

Optimization. We use AdamW with learning rate 10^{-3} for the perceptron and 10^{-2} for the curvature and the ideal point, weight decay 0.01, 1500 steps and margin $\gamma = 0.1$. The output dimension is $n = 128$ and the hidden width is 1024. We keep the checkpoint with the best pair accuracy on the held-out problems. Table 5 reports the held-out quality of all probes used in the main results.

Table 5: Held-out quality of the probes used in the main results. States come from the greedy solutions of 100 training problems, and 20 of these problems are held out for checkpoint selection. Pair accuracy is the fraction of held-out pairs with an entropy gap above 0.3 nats that the readout orders correctly, and it also selects the checkpoint. ρ is the Spearman correlation between the readout and the entropy on held-out states, and c is the learned curvature magnitude.

			Busemann			radius		
Model	Probe data	States	Pair acc.	ρ	c	Pair acc.	ρ	c
Qwen2.5-Math-1.5B	MATH	50,717	0.880	0.554	0.74	0.842	0.472	0.53
Qwen2.5-Math-1.5B	GSM8K	27,212	0.855	0.501	0.71	0.829	0.429	0.40
Qwen2.5-Math-1.5B	OlympiadBench	87,813	0.900	0.547	0.78	0.876	0.478	0.38
Llama-3.1-8B	MATH	41,873	0.903	0.747	0.85	0.894	0.711	0.72
Llama-3.1-8B	GSM8K	15,821	0.871	0.706	0.91	0.868	0.698	0.73
Qwen2.5-Math-7B	MATH	54,889	0.893	0.620	0.70	0.869	0.540	0.46
Qwen2.5-Math-7B	GSM8K	28,462	0.853	0.444	0.78	0.824	0.380	0.58

E Decoding Procedure

Algorithm 1 states the decoding loop. The cache rollback makes the edit act on the state whose next-token distribution is uncertain. On the greedy solutions of Qwen2.5-Math-1.5B-Instruct, the median entropy at hesitation points is 1.23 nats on MATH-500 and 1.44 nats on GSM8K, while one position later it is 0.075 and 0.048 nats. Without the rollback, the edit would land on a state at which the model is already confident and would leave the uncertain token unchanged.

Algorithm 1 HEST decoding for one sequence

Run the prompt through the model; obtain the cache C and the logits o for the first token
for $i = 1, \dots, T_{\max}$ **do**
  $y_i \leftarrow \arg\max o$; append y_i to the output; **if** y_i is the end token **then** stop
  $(o, C) \leftarrow M(y_i \mid C)$ {ordinary forward pass}
  if the entropy of $\mathrm{softmax}(o)$ exceeds θ **then**
    Remove the last position from C
    $h \leftarrow$ layer-ℓ state of y_i; $z \leftarrow E(h)$; $z' \leftarrow$ Eq. 6
    $\Delta h \leftarrow$ Eq. 7, clipped to $\|\Delta h\| \le \|h\|$
    $(o, C) \leftarrow M(y_i \mid C)$ with $h \leftarrow h + \Delta h$ at layer ℓ
  end if
end for
if the output reached $T_{\max} = 3072$ tokens without an end token, **return** the answer of greedy decoding

The edit is applied once per hesitation point, but its effect persists. The edited state is written into the keys and values of all later layers at that position, and every later token attends to it. The procedure is causal and needs no information about the rest of the generation. In a batch, the cache is rolled back for all sequences, and the edit is added only to the rows whose entropy exceeds the threshold. For the other rows, the recomputed forward pass is identical to the original one, so a hesitation point in any row costs one extra forward pass of the whole batch. The greedy answer for the fallback is needed only for the decodes that reach the limit.

F Grading Protocol

Grading free-form mathematical answers needs two steps, extraction and equivalence. We use one protocol for all models, because models differ in how they state their answers.

Extraction. We try four sources in order, namely the last $\texttt{\boxed\{\}}$ expression, an explicit answer statement such as "the answer is", a restatement of the final answer and the last number in the response. The model produces the restatement greedily, without steering and with at most 64 new tokens, when given its question, its reasoning and the prompt below.

Restatement prompt

```
Generate the final answer for the query {query} based on the
    reasoning process {reasoning} in this format: "Answer: \[
    \boxed{your_answer_here} \]". The entire answer should be
    contained completely within the \boxed{} command. Do not include
    any other text.
```

The Qwen2.5-Math models box their answers in 97 to 99% of the solutions. For Llama-3.1-8B-Instruct, the four sources are used in 31%, 6%, 59% and 4% of the MATH-500 solutions and in 0%, 8%, 90% and 2% of the GSM8K solutions. A grader that reads only $\texttt{\boxed\{\}}$ would therefore underestimate this model substantially.

Equivalence. The extracted answer is compared with the reference in both raw and normalized form, using a sequence of checks. Containers such as tuples, intervals and sets are compared element by element. Answers are then compared by string match and numerically, with integers compared exactly and decimals to a relative tolerance of 10^{-4}. The remaining cases go to the symbolic checker Math-Verify (Kydlíček, 2025) and, for scalars, to the equivalence check of the Qwen2.5-Math evaluation toolkit (Yang et al., 2024a). Each check runs under a time limit of 20 seconds.

Grading errors and paired comparisons. All arms are graded by the same protocol, and the restatement is produced greedily by the unsteered model, so the grade of a response depends only on its text. A solution whose text steering leaves unchanged therefore receives the same grade as under greedy decoding. Grading errors can affect the paired comparisons only through the changed solutions, since only a changed solution can be counted as fixed or broken in Appendix H.

G Calibration of the Gate Threshold

The rule aims at the same number of steered tokens per solution as on the development model. Its reference is the profile of Qwen2.5-Math-1.5B-Instruct at $\theta = 1$, the percentage of its steered decodes with 0 to 10, 11 to 24 and at least 25 steered tokens, which is $72/18/10$ on MATH-500 and $76/19/4$ on GSM8K. A new model has no steered decodes before θ is chosen. For a candidate θ, we therefore count the tokens with $H_i > \theta$ in each greedy solution of the evaluation set, obtained by one teacher-forced pass, and compute the same profile. We take the θ whose profile is closest to the reference in L_1 distance as the estimate (Table 6). On the development model, the greedy solutions give $73/19/9$ and $76/20/5$, within an L_1 distance of 3 of the reference, so the greedy count closely estimates the number of steered tokens. At the selected thresholds, the other models steer a median of 4 to 6 tokens per solution with the Busemann readout, against 5 for the development model. The rule uses no labels. It does read the unlabeled evaluation prompts and the model's own greedy solutions

Table 6: Gate-threshold calibration. For each candidate θ, the columns give the percentage of greedy tokens with $H > \theta$, the mean number of such tokens per solution, and the percentage of solutions with 0–10, 11–24 and at least 25 of them. L_1 is the distance to the reference profile of Qwen2.5-Math-1.5B-Instruct at $\theta = 1$, $72/18/10$ on MATH-500 and $76/19/4$ on GSM8K, which counts its steered tokens per solution (Appendix G). The selected threshold (bold) minimizes L_1. Entropies come from one teacher-forced pass over each model's greedy solutions and use no labels. For Qwen2.5-Math-7B on GSM8K, the 9 greedy solutions that reached the length limit are excluded.

θ	Tokens (%)	Per solution	Profile (%)	L_1
Llama-3.1-8B, MATH-500				
1.0	6.45	48.3	7/24/69	130
1.4	3.31	24.8	33/29/39	79
1.5	2.79	20.9	40/26/34	64
1.6	2.34	17.5	47/25/28	50
1.7	1.95	14.6	53/25/22	38
1.8	1.62	12.1	60/24/16	24
1.9	1.33	10.0	65/25/10	14
2.0	1.08	8.1	71/22/7	8
2.2	0.69	5.2	83/14/3	22
Llama-3.1-8B, GSM8K				
1.0	10.03	19.0	25/54/21	103
1.3	5.04	9.5	71/24/6	12
1.4	3.90	7.4	81/15/4	9
1.5	3.00	5.7	89/9/2	25
1.6	2.32	4.4	92/6/2	31
1.7	1.72	3.3	94/4/2	35
1.8	1.25	2.4	96/3/1	39
2.0	0.65	1.2	98/1/1	43

θ	Tokens (%)	Per solution	Profile (%)	L_1
Qwen2.5-Math-7B, MATH-500				
0.7	15.56	102.4	30/41/30	85
0.8	14.12	93.0	52/25/22	39
0.9	13.20	86.9	64/17/19	18
1.0	12.46	82.0	72/13/16	11
1.1	11.87	78.2	78/8/14	20
1.2	11.41	75.1	81/6/13	24
1.3	11.02	72.6	83/5/12	26
1.4	10.69	70.4	85/4/12	29
1.5	10.42	68.6	86/3/11	30
1.6	10.19	67.1	86/4/10	28
1.8	9.83	64.7	87/5/8	30
2.0	9.54	62.8	89/4/7	34
Qwen2.5-Math-7B, GSM8K				
0.7	3.94	12.0	59/35/6	35
0.8	2.66	8.1	81/16/3	9
0.9	2.00	6.1	88/9/2	24
1.0	1.57	4.8	93/6/1	33
1.1	1.25	3.8	95/4/1	37
1.2	1.01	3.1	96/3/1	39
1.3	0.84	2.6	97/2/1	41
1.4	0.72	2.2	98/1/1	43
1.5	0.61	1.9	98/1/1	43
1.6	0.54	1.6	98/1/1	43
1.8	0.43	1.3	99/1/1	44
2.0	0.37	1.1	99/0/1	45

to them, so the calibration adapts to the input distribution without seeing any answer. It matches counts per solution rather than rates per token, because solution lengths differ across models and datasets. For Llama-3.1-8B-Instruct, for example, the selected thresholds give 1.1% of the tokens on MATH-500 and 3.9% on GSM8K, but a similar number of hesitation points per solution.

H Full Results

Tables 7 and 8 give the number of problems fixed and broken relative to greedy decoding. They also give the number of problems for which the steered decode reached the length limit, so that the fallback was used. The fallback matters most for Llama-3.1-8B-Instruct on MATH-500, where greedy decoding itself reaches the length limit on 65 of the 500 problems. For the Busemann readout on this setting, 4 of the 17 fixed problems are problems on which greedy decoding reaches the length limit while the steered decode finishes with a correct answer. Without them, 13 problems are fixed and 11 broken. For Qwen2.5-Math-7B-Instruct on MATH-500, 16 greedy solutions degenerate into repetition, and 1 of the 8 fixed problems is among them.

Paired sign tests. Greedy decoding and HEST decode the same problems, so we compare them with a two-sided sign test on the problems whose outcome changes. As a sparse and conservative intervention, HEST changes the outcome of only 9 to 40 problems per setting, or 0.7 to 5.6% of a benchmark, which limits the power of the test in any single setting. On MATH-500, for example, 23 outcomes change, and $p < 0.05$ would require at least 17 of them in one direction, whereas HEST fixes 16 and breaks 7 ($p = 0.09$). In the order of Table 2, the six settings with the Busemann readout

give $p = 0.09$, 0.34, 0.34, 0.64, 0.39 and 0.51, and pooled over them HEST fixes 88 problems and breaks 63 ($p = 0.050$). This pooled test treats the six settings as independent, although the three models are evaluated on the same problems of each benchmark. The test gives clearer results for the comparisons that motivate the design of HEST. CAA, which edits every token, fixes 205 problems and breaks 369 over the six settings ($p < 10^{-11}$), so its losses are not due to chance. Against the Euclidean probe of Section 5 with the Busemann readout, HEST alone is right on 23 MATH-500 problems and the probe alone on 10 ($p = 0.035$). On GSM8K, the counts are 25 and 16 ($p = 0.21$), and pooled over the two benchmarks they are 48 and 26 ($p = 0.014$). These tests treat the probe as fixed, and since each setting uses a single trained probe, they do not measure the variation across training runs of the probe. Probes trained with the same recipe at layers 9 and 25 also improve MATH-500 accuracy (Appendix M), and the Busemann readout improves it at every step length of Table 1.

Table 7: Qwen2.5-Math-1.5B-Instruct, full results. Δ: change from greedy decoding in points. F/B: problems fixed (greedy wrong, HEST right) and broken (greedy right, HEST wrong). Cap: problems whose steered decode reached the 3072-token limit and fell back to the greedy answer. The largest change in each column is in bold.

		MATH-500			GSM8K			OlympiadBench		
Readout	t	Δ	F/B	Cap	Δ	F/B	Cap	Δ	F/B	Cap
Busemann	0.1	+0.60	7/4	6	−0.30	11/15	15	0.00	11/11	12
Busemann	0.25	+0.80	9/5	6	0.00	19/19	12	0.00	17/17	12
Busemann	0.5	**+1.80**	16/7	6	**+0.53**	23/16	14	−1.39	15/23	13
Busemann	1	+1.20	17/11	6	+0.45	31/25	8	−0.17	29/30	18
radius	0.1	0.00	6/6	2	−0.30	9/13	16	−0.35	8/10	13
radius	0.25	+1.00	12/7	7	−0.30	12/16	14	−0.52	11/14	17
radius	0.5	+0.20	11/10	4	−0.45	16/22	15	+0.87	18/13	15
radius	1	−0.80	11/15	4	−0.68	17/26	12	**+1.22**	26/19	16

Table 8: All models, full results for the two HEST configurations of Table 2. Columns as in Table 7, ℓ is the steered layer, and the best accuracy in each row is in bold.

					HEST, Busemann $t = 0.5$			HEST, radius $t = 0.25$		
Model	Dataset	ℓ	θ	Greedy	Accuracy	F/B	Cap	Accuracy	F/B	Cap
Qwen2.5-Math-1.5B	MATH-500	19	1.0	74.20	**76.00** (+1.80)	16/7	6	75.20 (+1.00)	12/7	7
	GSM8K	19	1.0	84.84	**85.37** (+0.53)	23/16	14	84.53 (−0.30)	12/16	14
Llama-3.1-8B	MATH-500	13	2.0	50.80	**52.00** (+1.20)	17/11	78	51.00 (+0.20)	12/11	64
	GSM8K	13	1.4	**84.61**	84.31 (−0.30)	18/22	2	84.53 (−0.08)	19/20	5
Qwen2.5-Math-7B	MATH-500	19	1.0	84.00	**84.80** (+0.80)	8/4	13	84.20 (+0.20)	5/4	14
	GSM8K	19	0.8	95.38	**95.60** (+0.23)	6/3	12	95.45 (+0.08)	2/1	9

I CAA Baseline

The CAA direction v is the raw difference of means of Section 4.1. Relative to the median norm of the hidden states, its norm is 0.33 and 0.24 for Qwen2.5-Math-1.5B-Instruct on MATH-500 and GSM8K, 0.30 and 0.23 for Llama-3.1-8B-Instruct, and 0.18 and 0.22 for Qwen2.5-Math-7B-Instruct. A single HEST edit has a median relative norm of 0.06 on the development model (Table 13), and it is applied to a small fraction of the tokens. Table 9 gives further statistics. Adding the fallback of HEST to CAA does not change the conclusion: CAA remains below greedy decoding in all six settings.

J Hesitation Groups and Hesitation Density

Table 10 gives the change in accuracy for each group of hesitation points and for the three HEST configurations. It also lists the greedy accuracy g, the fixed rate f and the broken rate b of each

Table 9: CAA baseline in detail. CAA adds the raw difference-of-means vector (multiplier 1) at every generated token and is reported without the fallback. Cap: decodes that reached the 3072-token limit. Length: mean solution length relative to greedy decoding. Changed: percentage of solutions whose text differs from greedy decoding. The last column applies the same fallback as HEST and is given only as a diagnostic.

Model	Dataset	Greedy	CAA	F/B	Cap	Length	Changed (%)	With fallback
Qwen2.5-Math-1.5B	MATH-500	74.20	70.00 (−4.20)	26/47	36	1.26	100	72.40 (−1.80)
Qwen2.5-Math-1.5B	GSM8K	84.84	84.23 (−0.61)	58/66	8	0.91	99	84.38 (−0.45)
Llama-3.1-8B	MATH-500	50.80	38.20 (−12.60)	31/94	131	1.33	100	44.20 (−6.60)
Llama-3.1-8B	GSM8K	84.61	80.06 (−4.55)	56/116	7	0.87	100	80.21 (−4.40)
Qwen2.5-Math-7B	MATH-500	84.00	82.60 (−1.40)	19/26	10	0.99	97	83.20 (−0.80)
Qwen2.5-Math-7B	GSM8K	95.38	95.00 (−0.38)	15/20	6	0.95	98	95.00 (−0.38)

group. The groups were fixed before the analysis. The group with at least 25 hesitation points could be split further. However, the part with at least 50 hesitation points holds only 2, 6 and 2 problems for Qwen2.5-Math-1.5B-Instruct on MATH-500 and GSM8K and for Llama-3.1-8B-Instruct on MATH-500.

Table 11 groups problems instead by hesitation density, which is the fraction of tokens in the greedy solution that exceed the threshold. These groups were chosen after the results of Table 10 were known. With the Busemann readout at $t = 0.5$, the densest group has the largest loss per problem for Qwen2.5-Math-1.5B-Instruct on GSM8K and OlympiadBench and for Llama-3.1-8B-Instruct on both benchmarks. This group holds only 20, 15, 2 and 60 problems in these settings. The median density of the greedy solutions of Qwen2.5-Math-1.5B-Instruct is 1.07% on MATH-500, 1.62% on GSM8K and 2.01% on OlympiadBench.

Table 10: Change from greedy decoding by the number of hesitation points in the greedy solution. For each group: Δ in points, the percentage of problems in the group (%), the greedy accuracy g, the fixed rate f among greedy-wrong problems and the broken rate b among greedy-right problems, so that $\Delta = 100\,[(1-g)f - gb]$ for the unrounded rates.

			0–10					11–24					≥ 25				
Model	Data	Setting	Δ	%	g	f	b	Δ	%	g	f	b	Δ	%	g	f	b
Qwen2.5-Math-1.5B	MATH-500	Busemann 0.5	+0.55	73	0.89	0.10	0.01	+5.32	19	0.41	0.15	0.08	+4.65	9	0.19	0.11	0.25
Qwen2.5-Math-1.5B	MATH-500	Busemann 0.25	−0.55	73	0.89	0.00	0.01	+2.13	19	0.41	0.07	0.05	+9.30	9	0.19	0.14	0.13
Qwen2.5-Math-1.5B	MATH-500	radius 0.25	−1.10	73	0.89	0.00	0.01	+3.19	19	0.41	0.09	0.05	+13.95	9	0.19	0.20	0.13
Qwen2.5-Math-1.5B	GSM8K	Busemann 0.5	+0.10	76	0.91	0.05	0.00	+1.16	20	0.72	0.16	0.05	+4.69	5	0.44	0.17	0.11
Qwen2.5-Math-1.5B	GSM8K	Busemann 0.25	−0.40	76	0.91	0.04	0.01	+0.78	20	0.72	0.12	0.04	+3.13	5	0.44	0.17	0.14
Qwen2.5-Math-1.5B	GSM8K	radius 0.25	−0.40	76	0.91	0.02	0.01	−0.39	20	0.72	0.08	0.04	+1.56	5	0.44	0.11	0.11
Qwen2.5-Math-1.5B	OlympiadBench	Busemann 0.5	+0.49	36	0.76	0.08	0.02	−4.09	30	0.35	0.05	0.22	−1.01	34	0.11	0.03	0.32
Qwen2.5-Math-1.5B	OlympiadBench	Busemann 0.25	−0.97	36	0.76	0.06	0.03	−0.58	30	0.35	0.06	0.13	+1.52	34	0.11	0.04	0.18
Qwen2.5-Math-1.5B	OlympiadBench	radius 0.25	0.00	36	0.76	0.02	0.01	0.00	30	0.35	0.05	0.08	−1.52	34	0.11	0.03	0.36
Llama-3.1-8B	MATH-500	Busemann 0.5	+0.84	71	0.66	0.08	0.03	+1.82	22	0.15	0.06	0.24	+2.94	7	0.06	0.03	0.00
Llama-3.1-8B	MATH-500	Busemann 0.25	0.00	71	0.66	0.08	0.04	0.00	22	0.15	0.09	0.47	0.00	7	0.06	0.00	0.00
Llama-3.1-8B	MATH-500	radius 0.25	−0.56	71	0.66	0.03	0.03	+2.73	22	0.15	0.09	0.29	0.00	7	0.06	0.00	0.00
Llama-3.1-8B	GSM8K	Busemann 0.5	−0.28	81	0.90	0.05	0.01	+1.00	15	0.67	0.13	0.05	−6.12	4	0.31	0.12	0.47
Llama-3.1-8B	GSM8K	Busemann 0.25	−0.09	81	0.90	0.07	0.01	−1.50	15	0.67	0.07	0.06	−2.04	4	0.31	0.09	0.27
Llama-3.1-8B	GSM8K	radius 0.25	+0.47	81	0.90	0.09	0.00	−1.00	15	0.67	0.12	0.08	−8.16	4	0.31	0.06	0.40
Qwen2.5-Math-7B	MATH-500	Busemann 0.5	0.00	72	0.97	0.08	0.00	0.00	13	0.78	0.14	0.04	+5.06	16	0.32	0.09	0.04
Qwen2.5-Math-7B	MATH-500	Busemann 0.25	+0.28	72	0.97	0.17	0.00	0.00	13	0.78	0.07	0.02	+3.80	16	0.32	0.07	0.04
Qwen2.5-Math-7B	MATH-500	radius 0.25	−0.28	72	0.97	0.00	0.00	−3.17	13	0.78	0.00	0.04	+5.06	16	0.32	0.09	0.04
Qwen2.5-Math-7B	GSM8K	Busemann 0.5	+0.28	80	0.98	0.12	0.00	−0.93	16	0.93	0.07	0.02	+4.26	4	0.55	0.10	0.00
Qwen2.5-Math-7B	GSM8K	Busemann 0.25	+0.09	80	0.98	0.04	0.00	−0.47	16	0.93	0.00	0.01	+2.13	4	0.55	0.05	0.00
Qwen2.5-Math-7B	GSM8K	radius 0.25	+0.09	80	0.98	0.04	0.00	0.00	16	0.93	0.00	0.00	0.00	4	0.55	0.05	0.04

K How Many Tokens HEST Steers

Table 12 describes how many tokens HEST steers per solution. The median fraction lies between 0.8 and 3.0% in every setting. Pooled over all tokens, the fraction is between 1.0 and 4.2% except for Qwen2.5-Math-7B-Instruct, where a few steered decodes that reach the length limit repeat

Table 11: Change from greedy decoding by hesitation density, the fraction of tokens in the greedy solution with $H > \theta$. Columns as in Table 10, except that the second column of each group is the number of problems. These groups were defined after the results of Table 10 were known.

			$< 2\%$					2–10%					$> 10\%$				
Model	Data	Setting	Δ	n	g	f	b	Δ	n	g	f	b	Δ	n	g	f	b
Qwen2.5-Math-1.5B	MATH-500	Busemann 0.5	+1.58	380	0.84	0.15	0.01	+2.50	120	0.44	0.10	0.08			–		
Qwen2.5-Math-1.5B	MATH-500	Busemann 0.25	0.00	380	0.84	0.03	0.01	+3.33	120	0.44	0.10	0.06			–		
Qwen2.5-Math-1.5B	MATH-500	radius 0.25	−0.26	380	0.84	0.06	0.02	+5.00	120	0.44	0.12	0.04			–		
Qwen2.5-Math-1.5B	GSM8K	Busemann 0.5	+0.26	764	0.92	0.05	0.00	+1.31	535	0.76	0.15	0.03	−10.00	20	0.55	0.11	0.27
Qwen2.5-Math-1.5B	GSM8K	Busemann 0.25	−0.13	764	0.92	0.03	0.00	+0.19	535	0.76	0.12	0.03	0.00	20	0.55	0.22	0.18
Qwen2.5-Math-1.5B	GSM8K	radius 0.25	−0.26	764	0.92	0.02	0.00	−0.37	535	0.76	0.08	0.03	0.00	20	0.55	0.00	0.00
Qwen2.5-Math-1.5B	OlympiadBench	Busemann 0.5	−0.35	287	0.61	0.07	0.05	−2.20	273	0.22	0.03	0.21	−6.67	15	0.13	0.00	0.50
Qwen2.5-Math-1.5B	OlympiadBench	Busemann 0.25	−0.35	287	0.61	0.08	0.06	0.00	273	0.22	0.03	0.11	+6.67	15	0.13	0.08	0.00
Qwen2.5-Math-1.5B	OlympiadBench	radius 0.25	+0.70	287	0.61	0.04	0.01	−1.83	273	0.22	0.03	0.18	0.00	15	0.13	0.08	0.50
Llama-3.1-8B	MATH-500	Busemann 0.5	+1.06	377	0.55	0.05	0.02	+2.48	121	0.36	0.12	0.14	−50.00	2	1.00	–	0.50
Llama-3.1-8B	MATH-500	Busemann 0.25	−0.27	377	0.55	0.04	0.04	+1.65	121	0.36	0.14	0.20	−50.00	2	1.00	–	0.50
Llama-3.1-8B	MATH-500	radius 0.25	−0.27	377	0.55	0.02	0.02	+2.48	121	0.36	0.10	0.11	−50.00	2	1.00	–	0.50
Llama-3.1-8B	GSM8K	Busemann 0.5	−0.53	375	0.92	0.00	0.01	−0.11	884	0.85	0.10	0.02	−1.67	60	0.37	0.11	0.23
Llama-3.1-8B	GSM8K	Busemann 0.25	−1.07	375	0.92	0.00	0.01	0.00	884	0.85	0.09	0.02	−1.67	60	0.37	0.08	0.18
Llama-3.1-8B	GSM8K	radius 0.25	+0.27	375	0.92	0.03	0.00	+0.23	884	0.85	0.12	0.02	−6.67	60	0.37	0.05	0.27
Qwen2.5-Math-7B	MATH-500	Busemann 0.5	0.00	362	0.95	0.11	0.01	+2.13	94	0.70	0.14	0.03	+4.55	44	0.23	0.06	0.00
Qwen2.5-Math-7B	MATH-500	Busemann 0.25	+0.55	362	0.95	0.17	0.00	+1.06	94	0.70	0.07	0.02	+2.27	44	0.23	0.06	0.10
Qwen2.5-Math-7B	MATH-500	radius 0.25	−0.55	362	0.95	0.00	0.01	+2.13	94	0.70	0.11	0.02	+2.27	44	0.23	0.06	0.10
Qwen2.5-Math-7B	GSM8K	Busemann 0.5	+0.15	652	0.98	0.08	0.00	0.00	638	0.95	0.09	0.00	+6.90	29	0.45	0.13	0.00
Qwen2.5-Math-7B	GSM8K	Busemann 0.25	0.00	652	0.98	0.00	0.00	0.00	638	0.95	0.03	0.00	+3.45	29	0.45	0.06	0.00
Qwen2.5-Math-7B	GSM8K	radius 0.25	0.00	652	0.98	0.00	0.00	+0.16	638	0.95	0.03	0.00	0.00	29	0.45	0.06	0.08

Table 12: How many tokens HEST steers (Busemann, $t = 0.5$). Per problem, the steered fraction is the number of steered tokens divided by the length of the steered solution. Pooled is the total number of steered tokens over the total number of generated tokens. Steered per solution: mean number of steered tokens. No steered token: percentage of solutions in which no token is steered. Capped: steered decodes that reached the 3072-token limit, and the percentage of all steered tokens that fall in them. A few capped decodes repeat themselves and pass the threshold many times, which inflates the pooled fraction for Qwen2.5-Math-7B. The median is not affected.

		Per problem (%)			Pooled (%)		Steered per	No steered	Capped	
Model	Dataset	Median	Mean	90th pct.	All	Excl. capped	solution	token (%)	n	Steered (%)
Qwen2.5-Math-1.5B	MATH-500	1.07	1.58	3.57	1.55	1.61	8.9	3.4	6	3
Qwen2.5-Math-1.5B	GSM8K	1.64	2.37	5.31	2.34	2.45	7.8	5.7	14	6
Qwen2.5-Math-1.5B	OlympiadBench	2.05	2.87	6.27	2.46	2.60	20.4	0.9	13	3
Llama-3.1-8B	MATH-500	0.80	1.32	3.20	1.01	1.42	8.3	17.4	78	41
Llama-3.1-8B	GSM8K	3.00	3.80	7.49	4.18	4.22	7.7	2.9	2	1
Qwen2.5-Math-7B	MATH-500	1.04	5.11	9.21	14.79	6.38	99.8	5.8	13	62
Qwen2.5-Math-7B	GSM8K	2.00	3.33	4.56	10.13	2.59	33.4	0.8	12	77

themselves and pass the threshold many times. These decodes contain 62% of all steered tokens on MATH-500 and 77% on GSM8K, and the fallback replaces their answers by those of greedy decoding.

L Diversity of the Steering Directions

We apply the HEST step to 400 states of Qwen2.5-Math-1.5B-Instruct at layer 19 whose entropy exceeds 1. The states come from 58 MATH training problems, and we compare the resulting edits Δh (Table 13). A fixed steering vector would give pairwise cosines of 1 and a single principal component. The HEST edits have a median pairwise cosine of 0.21 for the Busemann readout, and 32 principal components are needed to explain half of their variance. Even the mean edit has a median cosine of only 0.47 with the individual edits. The edits are more similar within a problem than across problems, with pairwise cosines of 0.34 within a problem and 0.21 across problems for the Busemann readout.

Table 13: Diversity of the HEST edits on 400 hesitation points of Qwen2.5-Math-1.5B-Instruct. The columns give the median pairwise cosine between edits, with the 10th and 90th percentiles in parentheses, the variance explained by the first principal component and by the first five, the number of components needed for half of the variance, the median relative size $\|\Delta h\|/\|h\|$, and the median cosine between each edit and the mean edit.

Readout	Pairwise cos.	PC 1	PCs 1–5	PCs for 50%	Size	Cos. to mean
Busemann, $t = 0.5$	0.210 (0.113, 0.334)	4.7%	17.6%	32	0.060	0.474
radius, $t = 0.25$	0.251 (0.153, 0.377)	5.6%	18.8%	27	0.017	0.510

M Ablation Details

Table 14 gives the full statistics of the ablations of Section 5. All variants are graded with the protocol of Appendix F. The fallback of Section 3.4 changes the accuracy of every variant of HEST by at most 0.40 points, so the conclusions do not depend on it.

Threshold and layer. The threshold $\theta = 1$ was fixed before the ablation (Section 3.5). Lowering it to $\ln 2$ nearly triples the median fraction of steered tokens and lowers accuracy by 1.00 point, and $\theta = 0$ is the every-token variant. Raising it to $\ln 4$ or $\ln 8$ steers at most 0.5% of the tokens, and only 11 and 4 problems change their outcome. The probes for layers 9 and 25 are trained with the same recipe on the same 100 training problems. They steer the same fraction of tokens as the probe for layer 19 and give gains of 0.60 and 0.20 points, against 1.80 points at layer 19.

Hesitation points and step boundaries. On the 500 greedy MATH-500 solutions, a solution has a median of 6 step boundaries and 5 hesitation points. Only 356 of the 3,974 step starts are hesitation points (9.0%), and only 356 of the 4,333 hesitation points are step starts (8.2%). A hesitation point lies a median of 27 tokens after the last step boundary. The two variants thus edit about as many tokens, but at different positions.

Fixed and broken solutions by direction. Table 15 splits the changed solutions by their greedy outcome. The break rates are low for all three directions. The fix rates differ more, and the probe direction has the highest fix rate on both benchmarks. Over the two benchmarks, the probe direction is correct and the fixed direction wrong on 39 problems, and the opposite holds on 33 problems. Against the reverse direction, the counts are 45 and 33.

N Euclidean Probes

The Euclidean probes of Section 5 take the limit $c \to 0$ of the probe of Section 3. They keep the perceptron, the loss of Eq. 5, the training states, the held-out problems, the optimizer and the number of steps of Appendix D. Only the curvature and the operations that depend on it change.

Formulas. As $c \to 0$, the map $\exp_0^c$ tends to the identity, and the probe becomes $E(h) = \phi\big((h - \mu)/\sigma\big)/\omega$. The conformal factor λ_z^c tends to 2, so the metric tends to $4g^E$ and the distance to $d_0(z, z') = 2\|z - z'\|$. The radius readout tends to $r_0(z) = 2\|z\|$, whose level sets are spheres around the origin. The Busemann readout satisfies $b(z)/(2\sqrt{c}) \to -\langle \xi, z\rangle$. We therefore use the linear readout $b_0(z) = -\langle \xi, z\rangle$ with a learned unit vector ξ, which is half the Busemann function of d_0 for the ideal point in the direction ξ. Its level sets are parallel hyperplanes orthogonal to ξ, so its gradient is the same at every point. In Eq. 6, Möbius addition tends to vector addition and $\tanh(\sqrt{c}\, t/2)/\sqrt{c}$ tends to $t/2$, so the step becomes

$$z' = z + \frac{t}{2}\, u. \tag{16}$$

This step moves z by $d_0(z, z') = t$, lowers r_0 by t when $t \le r_0(z)$ and lowers b_0 by $t/2$, which is the limit of Proposition 1. For the linear readout, $u = \xi$ at every state, so the step in the probe space is the same vector everywhere, and only the Jacobian of the probe makes the edit depend on the state. The Jacobian of Appendix C loses the factor of $\exp_0^c$. The mapping back to the hidden state, the clipping and the gate are those of HEST.

Table 14: Full statistics of the ablations on Qwen2.5-Math-1.5B-Instruct, Busemann readout, $t = 0.5$. Accuracy in % with the change from greedy decoding in points, with the fallback and without it. F/B: problems fixed and broken. Cap: steered decodes that reach the limit of 3072 tokens. Steered: median and pooled fraction of steered tokens in %. Edits: median number of steered tokens per solution. Step: median over problems of the mean $\|\Delta h\|/\|h\|$ of a solution. For HEST on MATH-500 it is measured at the hesitation points of the greedy solutions, and it was not logged for HEST on GSM8K. Length: median number of tokens, against 470 for greedy decoding on MATH-500 and 289 on GSM8K.

	Variant	Accuracy	F/B	Cap	No fallback	Steered	Edits	Step	Length
Positions MATH-500	Hesitation points (HEST)	76.00 (+1.80)	16/7	6	76.00	1.1 / 1.6	5	0.064	472
	Every token	74.00 (−0.20)	20/21	7	73.80	100 / 100	488	0.067	488
	Step boundaries	73.80 (−0.40)	5/7	6	73.80	1.4 / 1.4	6	0.072	480
Direction MATH-500	Fixed	76.00 (+1.80)	13/4	6	76.00	1.1 / 1.6	5	0.060	458
	Reverse	74.60 (+0.40)	9/7	7	74.20	1.1 / 1.5	5	0.068	468
Direction GSM8K	Probe (HEST)	85.37 (+0.53)	23/16	14	85.37	1.6 / 2.3	5	–	290
	Fixed	84.91 (+0.08)	20/19	11	84.91	1.6 / 2.4	5	0.060	290
	Reverse	84.99 (+0.15)	16/14	15	84.91	1.6 / 2.4	5	0.066	290
Threshold MATH-500	$\theta = \ln 2$	73.20 (−1.00)	10/15	5	73.00	3.1 / 3.8	15	0.067	476
	$\theta = 1$ (HEST)	76.00 (+1.80)	16/7	6	76.00	1.1 / 1.6	5	0.064	472
	$\theta = \ln 4$	74.40 (+0.20)	6/5	7	74.20	0.2 / 0.5	1	0.063	476
	$\theta = \ln 8$	74.60 (+0.40)	3/1	6	74.80	0.0 / 0.1	0	0.060	470
Layer MATH-500	$\ell = 9$	74.80 (+0.60)	10/7	3	74.80	1.1 / 1.6	5	0.052	474
	$\ell = 19$ (HEST)	76.00 (+1.80)	16/7	6	76.00	1.1 / 1.6	5	0.064	472
	$\ell = 25$	74.40 (+0.20)	12/11	6	74.00	1.1 / 1.6	5	0.069	470

Table 15: Changed solutions by direction, with the gate and the step size of HEST. Fixed: fraction of the greedy-wrong problems with a changed solution that the variant fixes. Broken: fraction of the greedy-right problems with a changed solution that it breaks.

	Fixed		Broken	
Direction	MATH-500	GSM8K	MATH-500	GSM8K
Probe (HEST)	0.154	0.204	0.042	0.038
Fixed	0.127	0.155	0.022	0.043
Reverse	0.088	0.130	0.044	0.037

Step length. A step of length t does not change the hidden state by the same amount in the two geometries, since the probes differ in their metric and in their scale. We therefore match the size of the edit instead of the step length. For the MATH-500 runs, we use the 400 hesitation points of Appendix L. For the GSM8K runs, we use all 300 hesitation points of the probe-training solutions, which come from 85 problems. On these states, we measure the median $\|\Delta h\|/\|h\|$ of HEST at its step length and find by bisection the Euclidean t that gives the same median, so no test label enters the choice (Table 16). On the MATH-500 test decodes, the median edit is 0.061 for the Euclidean Busemann probe and 0.066 for HEST. HEST with half the step length still gains 0.80 points (Table 1), so this difference of 8% does not explain the gap in accuracy. On the GSM8K test decodes, the median edit of the Euclidean Busemann probe is 0.058, as on the training states. The step length of HEST was chosen on the same test sets (Section 3.5), whereas the Euclidean step length only matches its edit size. On MATH-500, however, HEST gains accuracy at every step length of Table 1, by 0.60 to 1.80 points, while the Euclidean Busemann probe loses 0.80 points at the matched edit size.

Probe quality and results. The Euclidean probes order the held-out entropy pairs at least as well as the hyperbolic probes. Curvature thus brings no advantage in predicting the entropy, and the difference in steering lies in the direction that each geometry gives. On MATH-500, the gate steers a median of 1.1% of the tokens in all four runs. The Euclidean Busemann probe changes 245 solutions and HEST 270. Of the changed solutions that greedy decoding gets wrong, the Euclidean probe fixes 10.6% and HEST 15.4%. Of those that greedy decoding gets right, it breaks 9.3% and HEST 4.2%. The Euclidean Busemann edits are also more similar to each other. On the 400 hesitation points of Appendix L, their median pairwise cosine is 0.30, against 0.21 for HEST, and

their remaining variation spreads over more principal components, 41 against 32 for half of the variance. The radius edits of the two geometries are about equally diverse, with median cosines of 0.25 in both. On GSM8K, the gate steers a median of 1.6% of the tokens in all four runs. The Euclidean Busemann probe changes 498 solutions and HEST 536. Of the changed solutions that greedy decoding gets wrong, the Euclidean probe fixes 13.8% and HEST 20.4%, and of those that it gets right, the two break 4.4% and 3.8%. With the radius readout, the two geometries fix and break similar fractions and reach the same accuracy.

Table 16: Euclidean probes against the probes of HEST on Qwen2.5-Math-1.5B-Instruct, with $\theta = 1$ and layer 19. c: curvature magnitude, which is 0 for the Euclidean probes. Pair acc. and ρ: held-out quality as in Table 5. Step: median $\|\Delta h\|/\|h\|$ on the hesitation points of the probe-training solutions, which the step length t of the Euclidean probe matches. Accuracy in % with the change from greedy decoding in points, and the better accuracy of each pair is in bold. F/B: problems fixed and broken. Changed: solutions whose text differs from greedy decoding. Fix rate: fraction of the changed greedy-wrong solutions that become right. Break rate: fraction of the changed greedy-right solutions that become wrong. Paired: problems on which only HEST is right and on which only the Euclidean probe is right.

Readout	Geometry	c	Pair acc.	ρ	t	Step	Accuracy	F/B	Changed	Fix rate	Break rate	Paired
MATH-500, greedy accuracy 74.20												
Busemann	Hyperbolic (HEST)	0.74	0.880	0.554	0.5	0.060	**76.00** (+1.80)	16/7	270	0.154	0.042	–
	Euclidean	0	0.895	0.589	0.613	0.060	73.40 (−0.80)	10/14	245	0.106	0.093	23/10
radius	Hyperbolic (HEST)	0.53	0.842	0.472	0.25	0.017	**75.20** (+1.00)	12/7	181	0.156	0.067	–
	Euclidean	0	0.849	0.493	0.252	0.017	74.40 (+0.20)	6/5	185	0.077	0.047	12/8
GSM8K, greedy accuracy 84.84												
Busemann	Hyperbolic (HEST)	0.71	0.855	0.501	0.5	0.058	**85.37** (+0.53)	23/16	536	0.204	0.038	–
	Euclidean	0	0.880	0.633	0.672	0.058	84.69 (−0.15)	15/17	498	0.138	0.044	25/16
radius	Hyperbolic (HEST)	0.40	0.829	0.429	0.25	0.018	**84.53** (−0.30)	12/16	363	0.154	0.056	–
	Euclidean	0	0.832	0.470	0.241	0.018	**84.53** (−0.30)	11/15	354	0.143	0.054	10/10

O PROBE STATES ON REAL DATA

Figure 4 shows how the trained probes embed real hidden states of the development model on MATH-500, and Figure 5 shows all models and datasets. The states come from the 20 validation problems that were held out from probe training, and each panel shows 3,000 of them sampled at random.

How the states are drawn. The probe space has $n = 128$ dimensions, so any two-dimensional picture must drop information. We keep the information that the readouts use. Let $y = \sqrt{c}\, z$, which lies in the unit ball. For the Busemann probe, we draw y at distance $\|y\|$ from the center and at its angle to ξ, which lies between 0° and 180°. The Busemann readout depends only on these two quantities, because $\|\xi - y\|^2 = 1 - 2\langle \xi, y\rangle + \|y\|^2$. Each state therefore keeps its exact readout, and the dashed level sets are exact horocycles. Rotating a state about the axis through the origin and ξ changes neither quantity, so all states related by such a rotation are drawn at the same point, and a half disk is enough. A full disk would require a second direction to decide on which side of the axis each state falls, and this choice would be arbitrary. For the radius probe, the readout depends only on $\|y\|$, and we measure the angle from the first principal direction of the states.

Comparison with the schematic. The real states follow the same ordering as the schematic in Figure 1b,c, and the pattern is similar across models and datasets. They differ from the schematic in four ways. The states do not fill the ball, since $\|y\|$ lies between 0.39 and 0.82 for 90% of them, and almost no state lies at the origin or near the boundary. With the Busemann probe, confident and hesitant states differ more in their angle to ξ than in their norm. States with entropy below 0.1 nats have a median angle of 42° to 66°, and states above the threshold have a median angle of 66° to 87°. Confident states are thus closer to ξ in direction rather than in norm. With the radius probe, confident states lie at the inner edge of the occupied region rather than at the origin. Their median norm is 0.50 to 0.62, against 0.65 to 0.84 for states above the threshold. Finally, the two groups overlap. Across the 14 probes, the Spearman correlation between the readout and the entropy of the plotted states ranges from 0.39 to 0.76.

These differences do not affect the method. HEST uses only the gradient of the readout at the current state and a fixed step length. Proposition 1 holds wherever the state lies in the ball, so a step of length t lowers the readout by the same amount for every state. The schematic is drawn to make the ordering of the readout easy to see, and Figure 1 marks it as schematic.

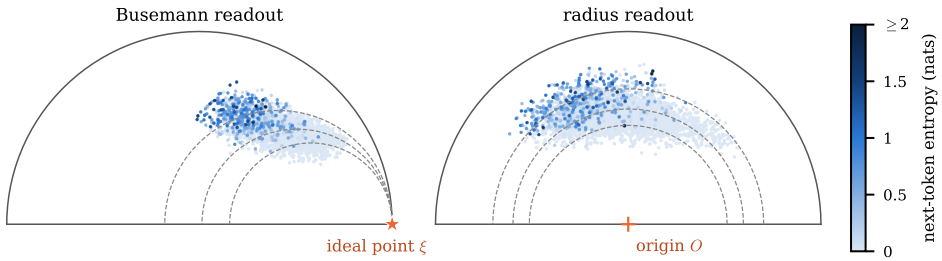


Figure 4: Held-out states of Qwen2.5-Math-1.5B-Instruct on MATH-500, embedded by the Busemann probe (left) and the radius probe (right). Each state $y = \sqrt{c}\, z$ is drawn at its norm and at its angle to the ideal point ξ (left) or to the first principal direction (right), which keeps its readout exact. Rotations about the horizontal axis leave the readout unchanged, so a half disk shows all states. Dashed curves are level sets of the readout at its 10th, 50th and 90th percentiles. Darker points have higher next-token entropy.

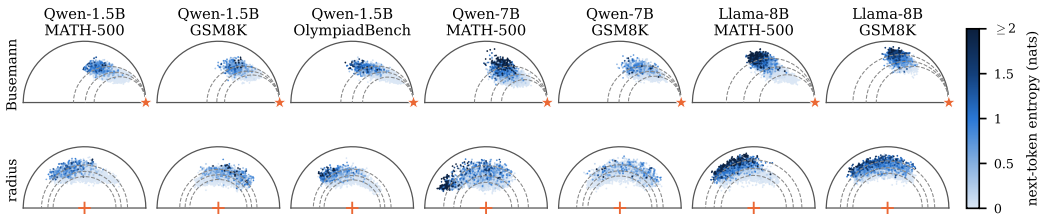


Figure 5: Held-out states of every model and dataset, drawn as in Figure 4. Top: Busemann probes. Bottom: radius probes.

P THE EXAMPLE OF SECTION 4.3 IN FULL

The problem is MATH-500 test problem 389 (geometry, level 3). Its diagram places the right angle at S and labels the side RT, the hypotenuse, with length 5. The reference answer is $\sqrt{21}/5$. We reproduce the problem and both solutions of Qwen2.5-Math-1.5B-Instruct verbatim. The steered solution sets the right angle correctly at S, but it then calls RS the hypotenuse and RT a leg. The two labels are swapped consistently, so the final ratio, and hence the answer, is correct even though these intermediate statements are not.

Problem

```
In $\triangle{RST}$, shown, $\sin{R}=\frac{2}{5}$.  What is $\sin{T}$?

[asy]
pair R,S,T;
T = (0,0);
S = (2,0);
R = (2,sqrt(21));
draw(R--S--T--R);
draw(rightanglemark(R,S,T,10));
label("$T$",T,SW);
label("$S$",S,SE);
```

```
label("$R$",R,NE);
label("$5$",(R+T)/2,NW);
[/asy]
```

Greedy decoding: 189 tokens, answer 1, wrong

```
To solve for \(\sin{T}\) in \(\triangle RST\), we start by analyzing
    the given information and the properties of the triangle.

Given:
- \(\sin{R} = \frac{2}{5}\)
- \(\triangle RST\) is a right triangle with \(\angle T = 90^\circ\)

Since \(\angle T\) is the right angle, \(\angle R\) and \(\angle S\)
    are complementary angles. Therefore, we have:
\[
\angle R + \angle S = 90^\circ
\]

We need to find \(\sin{T}\). However, since \(\angle T = 90^\circ\),
    we know that:
\[
\sin{90^\circ} = 1
\]

Thus, the value of \(\sin{T}\) is:
\[
\boxed{1}
\]
```

HEST, Busemann readout, $t = 0.5$: 583 tokens, 12 steered, answer $\sqrt{21}/5$, correct

```
To solve for \(\sin{T}\) in \(\triangle RST\), we start by analyzing
    the given information and the properties of the triangle.

Given:
- \(\sin{R} = \frac{2}{5}\)
- \(\triangle RST\) is a right triangle with \(\angle RST = 90^\circ\)

Since \(\angle RST = 90^\circ\), \(\angle R\) and \(\angle T\) are
    complementary angles. Therefore, we have:
\[
\angle R + \angle T = 90^\circ
\]

We know that the sine of an angle in a right triangle is the ratio of
    the length of the opposite side to the hypotenuse. For \(\angle
    R\), the opposite side is \(ST\) and the hypotenuse is \(RS\).
    Given \(\sin{R} = \frac{2}{5}\), we can write:
\[
\sin{R} = \frac{ST}{RS} = \frac{2}{5}
\]

Let \(ST = 2k\) and \(RS = 5k\) for some positive constant \(k\).

Next, we use the Pythagorean theorem to find the length of the
    remaining side \(RT\):
\[
RT^2 + ST^2 = RS^2
\]
\[
RT^2 + (2k)^2 = (5k)^2
\]
```

```
\[
RT^2 + 4k^2 = 25k^2
\]
\[
RT^2 = 25k^2 - 4k^2
\]
\[
RT^2 = 21k^2
\]
\[
RT = \sqrt{21k^2} = k\sqrt{21}
\]

Now, we need to find \(\sin{T}\). Since \(\angle T\) is complementary
    to \(\angle R\), we have:
\[
\sin{T} = \cos{R}
\]

The cosine of an angle in a right triangle is the ratio of the length
    of the adjacent side to the hypotenuse. For \(\angle R\), the
    adjacent side is \(RT\) and the hypotenuse is \(RS\). Therefore:
\[
\cos{R} = \frac{RT}{RS} = \frac{k\sqrt{21}}{5k} = \frac{\sqrt{21}}{5}
\]

Thus, we have:
\[
\sin{T} = \cos{R} = \frac{\sqrt{21}}{5}
\]

Therefore, the value of \(\sin{T}\) is:
\[
\boxed{\frac{\sqrt{21}}{5}}
\]
```