VISUAL REASONING WITH CHAIN-OF-COORDINATES

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce Chain-of-Coordinates (CoC), a representation for visual reasoning in which vision-language models (VLMs) autoregressively generate sequences of image coordinates. We provide a theoretical analysis of CoC, covering its representability, learnability, geometric approximation error, and computational advantage over direct answering on a family of connectivity problems. We introduce a two-stage reinforcement learning (RL) framework with a Fréchet-based geometric reward. Across six tasks, CoC improves curve tracing and consistently outperforms direct-answer training on reasoning tasks, with further gains from RL. Additional analyses show that CoC is most beneficial when connectivity cannot be inferred from direct visual cues, while the effectiveness of CoC prompting depends on a model's underlying tracing ability. Together, these results establish CoC as a paradigm for connecting visual perception and reasoning through explicit coordinate sequences.

1 INTRODUCTION

Spatial reasoning requires integrating local visual evidence into a coherent representation of geometric and topological structure. This remains challenging for vision-language models (VLMs): benchmarks such as BabyVision (Chen et al., 2026) and BlindTest (Rahmanzadehgervi et al., 2024) reveal substantial limitations in fine-grained visual perception and spatial reasoning. In particular, tasks such as line tracing and path following are effortless for humans yet remain difficult for VLMs.

We introduce *Chain-of-Coordinates (CoC)* as a common formulation for visual reasoning through ordered sequences of image coordinates. Given an image and a query, the VLM autoregressively generates a coordinate sequence conditioned on preceding coordinates. The resulting sequence maps the traced curve into the textual space, with consecutive coordinates defining a polyline approximation. CoC uses standard autoregressive text generation without architectural modifications, external tools, or intermediate image generation. It may serve directly as the desired output, such as a drawing or character stroke for computer use, or as a reasoning trace that supports a final answer (Figure 1).

Previous work on line tracing in VLMs has been primarily empirical (Panagopoulou et al., 2025; Collins et al., 2026), leaving its underlying properties poorly understood. We therefore investigate the following research questions: **RQ1.** Can a VLM generate coordinate sequences that represent curves in an image? **RQ2.** Can a VLM be trained to learn tracing rules that generalize to unseen curves? **RQ3.** How accurately do generated coordinates approximate the underlying geometry? **RQ4.** Does CoC provide a computational advantage over direct answering?

To address these questions, we provide a theoretical analysis of CoC's representability, learnability, geometric approximation, and computational expressivity. We construct a transformer-based VLM for curve tracing and establish sufficient conditions for generalization to unseen curves and lengths. We show that training a linear prediction head on all admissible local states yields exact tracing within the context limits. We derive reconstruction error bounds in Fréchet distance that account for sampling, quantization, and prediction errors. Last, we identify a family of endpoint-matching problems for which CoC has a conditional computational advantage over direct answering.

We propose a two-stage reinforcement learning (RL) based training framework to train VLMs to generate and reason with CoC. We introduce a geometric reward based on continuous Fréchet distance that evaluates complete traces across different samplings. This reward complements format and answer rewards to optimize both tracing fidelity and task correctness. Experiments across six tasks show improved curve tracing and visual reasoning, with CoC supervised fine-tuning (SFT)

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.



There are three animal patterns at the top of the picture and three environment patterns below, which are connected by lines in order. What is the connection method? (Assume the animal patterns above are A, B, C, and the environment patterns below are 1, 2, 3. The answer format is like: A-1, B-2, C-3)

A: ... <CoC>(119,374) (129,414) (139,449) (159,490) ... (614,750)</CoC> ...

B: ... <CoC>(532,403) (536,451) (546,490) (564,526) ... (973,777)</CoC> ...

C: ... <CoC>(872,396) (854,437) (841,480) (829,524) ... (254,818)</CoC> ...

<answer>A-2, B-3, C-1</answer>

Figure 1: **Visual reasoning with Chain-of-Coordinates.** An example from BabyVision Connect the lines (Chen et al., 2026): tracing each animal's line to its endpoint gives A–2, B–3, C–1. Colors and point labels are for visualization only, not additional model inputs.

outperforming direct-answer SFT on reasoning tasks. RL yields further gains, enabling a 2B VLM to match leading models such as GPT-5.6 Sol on the selected tasks. We also examine when CoC helps: explicit tracing is most useful when appearance cues do not reveal connectivity, while CoC prompting benefits depend on the model's underlying tracing ability.

Our main contributions are:

- We introduce CoC for representing curves and maintaining spatial state through autoregressive coordinate sequences. We provide a theoretical analysis of its representability, learnability, geometric approximation, and computational expressivity (Sections 3 and 4).
- We develop a two-stage RL-based training framework with a geometric reward based on Fréchet distance that evaluates curves across different coordinate samplings. Combined with format and answer rewards, it optimizes geometric fidelity and task correctness for both coordinate outputs and visual reasoning (Section 5.2).
- Experiments across six tasks demonstrate improved curve tracing and visual reasoning, with CoC training outperforming direct-answer SFT on reasoning tasks and RL yielding further gains. We also examine when appearance cues reduce the need for tracing and how prompting benefits relate to the model's underlying curve-tracing ability (Section 5.3).

2 Related Work

Visual Chain-of-Thought. Chain-of-thought (CoT) prompting uses intermediate textual steps to support reasoning (Wei et al., 2022). Visual CoT extends this approach with visual intermediate representations. Prior methods crop or zoom into relevant regions and condition subsequent reasoning on these local observations (Wu & Xie, 2024; Shao et al., 2024a; Jiang et al., 2025; Zheng et al., 2026), or construct annotations, sketches, and diagrams as intermediate reasoning steps (Hu et al., 2024; Menon et al., 2024; Wu et al., 2025; Collins et al., 2026), generate image tokens to visualize reasoning states (Li et al., 2025), or interleave latent visual tokens with text (Yang et al., 2026b). In contrast, CoC represents intermediate reasoning as textual coordinate sequences.

Coordinate sequences in VLMs. Coordinate sequences have been used for pixel-space navigation (Fu et al., 2026), segmentation (Song et al., 2026), route tracing on maps (Panagopoulou et al., 2025), and visual tracking (Liu & Lee, 2026). Existing methods combine textual reasoning with points and boxes (Lu et al., 2026) or use visual trajectories to support spatial understanding (Yang et al., 2026a). Other studies investigate visual path tracing and traversal in VLMs (Petrova et al., 2026; Hong et al., 2026). These works are primarily empirical; in contrast, we provide a theoretical analysis of CoC, characterizing when coordinate-based reasoning is representable and learnable, and when it offers a computational advantage. More broadly, we establish CoC as a general paradigm for maintaining spatial state in VLMs, and show that it enables reliable solutions to a family of endpoint-matching problems that remain challenging for existing approaches.

3 CHAIN-OF-COORDINATES

Spatial reasoning tasks. Given an image I and a query q, we study spatial reasoning problems that require locating, tracing, or reasoning about connectivity, correspondence, and containment over visual structures formed by curves. Representative examples are image-conditioned drawing and writing, Connect the lines, nonsemantic tracing, and nested-region reasoning (Figure 2).

Chain-of-Coordinates (CoC). Given an image I and a query q, a VLM with parameters θ generates a CoC as an ordered sequence of coordinates:

$$P = (p_0, \ldots, p_{K-1}), \qquad p_t = F_\theta(I, q, (p_0, \ldots, p_{t-1})), \quad 0 \le t < K. \tag{1}$$

The sequence contains $K \ge 0$ points, with $p_t = (x_t, y_t) \in \Omega = [0, 1]^2$ giving normalized horizontal and vertical coordinates. Connecting consecutive points with straight segments gives a polyline approximation of the curve. For point localization tasks, $K = 1$ and the CoC contains a single point (x_0, y_0). For curve tracing tasks, $K > 1$ and $p_0, \ldots, p_{K-1}$ are the predicted points. Multiple curves are represented by separate CoCs.

CoC uses textual coordinate tokens to record intermediate spatial states. Unlike Visual CoT methods that crop or annotate images and feed them back to the model, CoC keeps the original image fixed during inference. It invokes no external tools and generates no intermediate images; subsequent predictions condition on the input image and preceding generated text. Coordinates are represented as text at a specified precision, such as `(0.123,0.456)` with three decimal places. The tokenization of numbers varies across models: `123` corresponds to a single token in GPT-5, but to three tokens, `1`, `2`, and `3`, in Qwen3.5. Further details on coordinate representation are provided in Appendix A.1.

4 THEORETICAL ANALYSIS

We study how CoC represents visual structure and supports reasoning. We first establish that a transformer-based VLM can trace a path using a local rule (Section 4.1), then characterize when learning this rule enables generalization to unseen paths (Section 4.2). We next examine how sampling, coordinate rounding, and prediction errors affect reconstructed curves (Section 4.3). Finally, we study computational expressivity through endpoint matching, proving a separation between CoC and direct answering (Section 4.4).

4.1 REPRESENTABILITY

We take a two-dimensional binary image as input and construct a simple transformer-based VLM with a vision encoder and an autoregressive decoder. The encoder records each image patch's binary pixel values and position. The decoder traces a path using these features and previously generated coordinates. The transformer has a fixed number of layers, independent of image size and path length, while model size and context length may grow polynomially with the grid size (Appendix A).

Let $M \subseteq \Lambda_n = \{0, \ldots, n-1\}^2$ be the occupied pixels of an $n \times n$ image, $n \ge 1$. Consider a path $\bar{p}_0, \ldots, \bar{p}_T$ of distinct pixels in M. Barred coordinates are pixel indices; dividing by $\max\{1, n-1\}$ gives the normalized CoC. Choose a fixed, finite, nonempty set $\mathcal{D}$ of nonzero pixel moves, including each move's reverse (e.g. the eight neighbors). We require that a path point's only occupied neighbors are its predecessor and successor:

$$M \cap (\bar{p}_t + \mathcal{D}) = \{\bar{p}_j : 0 \le j \le T,\ |j - t| = 1\}, \qquad 0 \le t \le T. \tag{2}$$

For each move $d \in \mathcal{D}$, record whether the neighbor is occupied, $m_t(d)$, and whether it is the previous point, $b_t(d)$:

$$m_t(d) = \mathbf{1}[\bar{p}_t + d \in M], \qquad b_t(d) = \mathbf{1}[t > 0,\ \bar{p}_t + d = \bar{p}_{t-1}], \quad d \in \mathcal{D}, \qquad b_0(d) = 0. \tag{3}$$

Throughout the tracing constructions in this section and Section 4.2, the query supplies the start, coordinates are written exactly (Lemma 1), and context holds the query and full output (Appendix A). Output lengths are bounded by a polynomial in grid size; simple paths use at most n^2 records. Decoding *recovers* a trace if it emits $\bar{p}_0, \ldots, \bar{p}_T$ in order and then `EOS`.

Lemma 1 (Local information)**.** *The transformer computes m_t, b_t and can write the start or any selected in-grid neighbor of $\bar{p}_t$ as the next coordinate.*

Proof sketch. Attention reads the generated coordinates. Position comparisons identify the occupied neighbors and the previous point, and a lookup writes the selected coordinate in the format of Appendix A.1. The full proof is given in Appendix A.2. □

The tracing rule chooses the occupied neighbor other than the previous point, stopping with `EOS` if none remains. For output classes $\mathcal{A} = \mathcal{D} \cup \{\texttt{EOS}\}$, use the scores

$$s_t(d) = 2\big(m_t(d) - b_t(d)\big), \qquad s_t(\texttt{EOS}) = 1. \tag{4}$$

Theorem 2 (Exact local coordinate tracing)**.** *For every path satisfying Equation* (2)*, greedy decoding with scores* (4) *recovers the path, with margin at least* 1 *at every step.*

Proof. Before the endpoint, $m_t - b_t$ is one only in the next-point direction, whose score is 2; other moves score 0, and stopping scores 1. At the endpoint, all moves score 0, so the model stops. Induction from the supplied start gives the full path and termination. □

For thick strokes, the result requires a sampled centerline satisfying Equation (2). Section 4.2 extends the argument to crossings and junctions resolved by additional local information; Appendix A.3 covers other discrete outputs.

4.2 Learnability

We train only a linear head U, keeping the rest of the VLM constructed in Section 4.1 fixed. The head takes the decoder's local features: m_t records the occupied neighbors, and b_t identifies the previous point:

$$h_t = (m_t, b_t, 1), \qquad s_t = U h_t. \tag{5}$$

Here h_t is the local feature vector and s_t contains logits for each allowed direction and `EOS`; the constant 1 accounts for the bias. Feature computation is exact. Training uses teacher forcing: features are computed from ground-truth coordinate prefixes, with the next direction or `EOS` as the target.

The finite set of local feature vectors makes length generalization possible. With eight neighboring directions, only 49 occur under Equation (2), and paths of at most three points can cover them all (Appendix B.2). Training on these short paths can therefore cover the local predictions needed to decode longer paths.

Theorem 3 (Learning and length generalization)**.** *Suppose training covers every local feature vector occurring under Equation* (2)*. From any initialization, gradient descent on summed cross-entropy with step size $0 < \alpha \leq 1/\beta$ (Appendix B.1) eventually ranks the correct class first on all such vectors. The learned head then recovers every path satisfying Equation* (2)*.*

Proof sketch. Equation (4) gives weights for a linear head that predict every target label correctly. Lemma B.1 ensures that cross-entropy training eventually gives correct predictions on all covered feature vectors. Applying the learned head autoregressively recovers the full path. The full proof is given in Appendix B.1. □

The explicit linear head also tolerates small perturbations to its input features.

Corollary 4 (Robustness to feature perturbations)**.** *Under Theorem 2, if every entry of m_t and b_t is perturbed by at most $\varepsilon < 1/4$ along the path, greedy decoding with the explicit scores* (4) *still recovers it.*

Proof. Each move logit changes by at most $4\varepsilon < 1$. The logit for the next move therefore remains above the `EOS` logit of 1, and every other move remains below it. At the endpoint, all move logits are below 1, so decoding terminates. □

Appendix B.3 gives the learned head's robustness bound.

Proposition 5 (Learning with local context)**.** *Suppose a fixed procedure computes each step's local state from a fixed-size patch over a finite pixel alphabet and a bounded number of recent coordinates. The states take values in a finite set $\mathcal{S}$ over the whole family and determine the next move or* `EOS`*. If training covers $\mathcal{S}$, gradient descent as above on one-hot states yields a head that recovers every trace in the family.*

Proof sketch. One-hot encoding of the local states makes their target labels learnable by a linear head. Once training covers every state and the head predicts all targets correctly, autoregressive decoding recovers the full trace. The full proof is given in Appendix B.4. □

In line tracing, for example, recent coordinates and local image geometry can identify the smoothest continuation at a crossing. When this uniquely determines the correct branch, Proposition 5 guarantees that the model can learn the rule from examples covering all local states.

4.3 Geometric Approximation

We bound curve reconstruction error for tracing and drawing, then derive consequences for region masks, trace containment, and learned tracing.

Curve reconstruction. CoCs in this subsection are nonempty. Joining successive points of P with straight segments gives $\Pi(P)$; a single point gives a constant curve. We measure error with Fréchet distance, d_{F}, which respects traversal order (Appendix C.1).

For integer $B \geq 1$, coordinate quantization rounds p to the nearest grid point $Q_B(p)$ with spacing $1/B$, giving error at most $\eta_B = \sqrt{2}/(2B)$. The next lemma extends coordinate-error bounds to the connecting segments.

Lemma 6 (Stability under coordinate errors)**.** *For CoCs P, Q of equal length, if $\|p_i - q_i\|_2 \leq \delta$ for all i, then $d_{\mathrm{F}}(\Pi(P), \Pi(Q)) \leq \delta$.*

Proof. Along corresponding segments, the difference is a weighted average of the endpoint differences, so its length is at most δ. The single-point case follows directly. □

The next result adds the error from linear interpolation between curve samples.

Proposition 7 (Curve approximation)**.** *Let $\gamma : [0, L] \to \Omega$ be a unit-speed C^2 curve, $L > 0$, with curvature at most κ. Sample $p_0, \ldots, p_J$ in order, including both endpoints, with arc-length spacing at most h. If a CoC $\widehat{P} = (\widehat{p}_0, \ldots, \widehat{p}_J)$ satisfies $\|\widehat{p}_i - Q_B(p_i)\|_2 \leq e$ for all i, then*

$$d_{\mathrm{F}}(\gamma, \Pi(\widehat{P})) \leq \kappa h^2/8 + \eta_B + e =: \delta. \tag{6}$$

Proof sketch. Connecting exact samples introduces error at most $\kappa h^2/8$. Lemma 6 adds the rounding and prediction errors, $\eta_B + e$. The full proof is given in Appendix C.2. □

The terms bound interpolation, coordinate quantization, and prediction error. Halving the sample spacing reduces the interpolation term fourfold. The bound covers curves with corners if every corner is sampled and each smooth piece satisfies the curvature condition.

Mask reconstruction. Filling a closed CoC produces a region mask. The next result confines labeling errors to a band around the reference boundary.

Corollary 8 (Masks from boundary traces)**.** *Under Proposition 7, let γ be a simple closed curve and $\widehat{P}$ closed. The interior of γ and the even–odd fill of $\Pi(\widehat{P})$ differ only within distance δ of γ, even if $\Pi(\widehat{P})$ self-intersects.*

Proof sketch. Corresponding curve points are within δ. Moving between them along straight segments cannot change the inside/outside label of a point farther than δ from the reference boundary. The full proof is given in Appendix C.3. □

For multiple contours, each matched pair must satisfy the same conditions, with no missing or extra contours (Appendix C.3).

Trace containment. Let $R \subseteq \Omega$ contain the reference curve, for example the image region occupied by a thick stroke. Its clearance ρ is the distance from the reference curve to image points outside R (Appendix C.4).

Corollary 9 (Trace containment)**.** *Under Proposition 7, the entire predicted curve stays inside R whenever $\delta < \rho$.*

Proof. The Fréchet bound places every point of the predicted curve closer than ρ to some reference point. A point outside R would contradict the definition of ρ. □

Learned tracing. Combining Section 4.2 with the approximation bound leaves only interpolation and coordinate quantization error when the rounded samples are recovered exactly.

Corollary 10 (Accuracy after learning)**.** *Under Theorem 3, if the rounded samples from Proposition 7, written as grid indices, form a path satisfying Equation* (2)*, the learned head recovers them. Hence $e = 0$ and $d_{\mathrm{F}}(\gamma, \Pi(\widehat{P})) \le \kappa h^2/8 + \eta_B$.*

Proof. Theorem 3 recovers the integer-grid path. Normalizing its coordinates gives the rounded samples, so Proposition 7 gives the bound with $e = 0$. □

4.4 Computational Expressivity

We examine whether generating intermediate coordinates increases a VLM's computational expressivity. Under the computational assumptions below, a VLM using CoC solves a structured connectivity task by following a curve one block at a time, whereas direct answering cannot solve every input with the same bound on computation per token.

Endpoint matching. We use images formed by joining fixed-size blocks from left to right. Each block rearranges the curves among a fixed set of lanes using fixed polygonal templates with rational vertices, and marked crossings keep intersecting curves separate. Image size grows linearly with the number of blocks. Details are given in Appendix D.1.

Definition 11 (Visual endpoint connectivity)**.** For this image family, determine whether the two marked endpoints lie on the same curve.

Computation model. For an image of N_{img} bits, generating each token, including all image processing, must be computable by a polynomial-size, constant-depth Boolean circuit with majority gates (the class TC^0). Constant depth means a fixed number of sequential gate layers, independent of image and generated sequence length. We consider autoregressive models with deterministic generation over a fixed finite vocabulary. Each token depends only on the image and previously generated tokens. The class AC^0 omits majority gates; NC^1 allows logarithmic depth with gates taking a fixed number of inputs. Details are given in Appendix D.2.

CoC construction. The constructed model starts at the marked source and generates coordinates along the curve through each block. A boundary coordinate identifies the current lane, so this coordinate and the next block determine the continuation. The last coordinate gives the answer: it identifies either the marked target or a different endpoint. Thus tracing answers both connected and disconnected cases. Appendix D.2 specifies the coordinate sequence and its exact encoding.

Theorem 12 (CoC versus direct answering)**.** *Endpoint connectivity in this family is NC^1-complete under AC^0 reductions. Under the computation model above:*

- *(i) If $\mathsf{TC}^0 \neq \mathsf{NC}^1$, no model solves every image using $O(\log N_{\mathrm{img}})$ intermediate tokens; in particular, direct answering fails.*
- *(ii) A transformer family as in Section 4.1, with AC^0 per-token computation, solves every image by emitting the source curve's trace and the answer, using $O(N_{\mathrm{img}} \log N_{\mathrm{img}})$ tokens.*

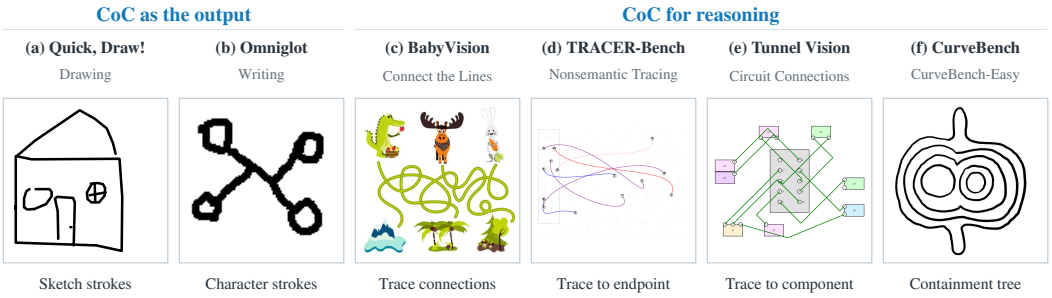


Figure 2: Task examples from (a) Quick, Draw! (Drawing) (Google Creative Lab, 2017); (b) Omniglot (Writing) (Lake et al., 2015); (c) BabyVision (Connect the Lines) (Chen et al., 2026); (d) TRACER-Bench (Nonsemantic Tracing, LT) (Guo et al., 2026); (e) Tunnel Vision (Circuit Connections, Single Color) (Berman & Deng, 2025); and (f) CurveBench-Easy (Mohseni et al., 2026).

Proof sketch. The block templates encode an NC^1-complete permutation-composition problem (Barrington, 1989; Li et al., 2024). With only logarithmically many intermediate tokens and a fixed vocabulary, there are polynomially many candidate output sequences. Checking them in parallel would put the task in TC^0, contradicting the assumption. CoC instead updates the current lane one block at a time. The full proof is given in Appendix D.3. □

Direct answering uses no intermediate tokens and is covered by (i). CoC meets the same per-token bound because $\mathsf{AC}^0 \subseteq \mathsf{TC}^0$; its advantage is that later steps can use positions computed earlier.

Proposition 13 (Parallel verification of CoC)**.** *Checking whether a candidate CoC in the format of Theorem 12(ii) is the source curve's trace, and whether a proposed answer matches it, is in* AC^0.

Proof sketch. Given the candidate trace, every step and the final answer can be checked independently in parallel. The full proof is given in Appendix D.4. □

5 Experiments

We select six tasks from existing benchmark datasets that are easy for humans yet challenging for VLMs, and for which explicit curve tracing with CoC potentially improves performance (Figure 2). Drawing and writing evaluate **CoC as the output**, measuring tracing quality across varied stroke patterns. Connectivity and containment tasks require **CoC for reasoning** to answer questions about endpoint matching and nested structures.

5.1 Data Collection

Quick, Draw! and Omniglot provide temporally ordered pen strokes as sampled (x, y) coordinates. We render the strokes as input images and use their coordinates as target CoC, simplified with Ramer–Douglas–Peucker at a tolerance of 5 units on the $[0, 1000]^2$ coordinate grid to shorten targets while preserving shape. We use class-level splits with ∼18k training examples per dataset and 50 test examples from held-out categories and character classes, respectively.

Synthetic data generation. For tasks that use CoC for reasoning, no benchmark provides trace-annotated training data, so we train on generated scenes and evaluate on the official test sets: for Circuit Connections we use the benchmark's own generator (~17k examples); for CurveBench, TRACER-Bench, and BabyVision, we implement a generator with random curves approximating their visual style (~10k examples each).

Each training example pairs an image and a task instruction with CoC in `<CoC>...</CoC>` tags; reasoning problems append an answer inside `<answer>...</answer>`.

Table 1: Model comparison across six datasets. Quick, Draw! and Omniglot use normalized Fréchet distance (lower is better). BabyVision (Connect the lines) and Tunnel Vision use answer accuracy; TRACER-Bench (LT) uses exact-match accuracy; CurveBench uses rooted-tree accuracy (0–1, higher is better). Direct answer SFT is the same as CoC SFT for Quick, Draw! and Omniglot. Closed-source models are evaluated using their default reasoning settings.

	CoC as the output		**CoC for reasoning**			
Model	Quick, Draw! ↓	Omniglot ↓	BabyVision ↑	TRACER-Bench ↑	Tunnel Vision ↑	CurveBench ↑
GPT-5.6 Sol	<u>0.289</u>	<u>0.151</u>	0.000	<u>0.980</u>	<u>0.552</u>	<u>0.800</u>
GPT-5.6 Luna	0.295	0.240	0.000	0.940	0.448	0.533
Gemini 3.8 Flash	0.393	0.202	<u>0.053</u>	<u>0.980</u>	0.480	0.711
Gemini 3.1 Pro	0.530	0.265	<u>0.053</u>	0.920	0.416	0.667
Claude Opus 5	0.590	0.558	<u>0.053</u>	0.960	0.336	0.444
Qwen3.5 2B (base)	0.973	0.917	0.000	0.300	0.160	0.022
+ Direct-answer SFT	—	—	0.000	0.840	0.752	0.467
+ CoC SFT	$0.266^{-0.707}$	$0.135^{-0.782}$	$0.053^{+0.053}$	$0.920^{+0.620}$	$0.936^{+0.776}$	$0.756^{+0.734}$
+ CoC SFT + RL	$\underline{0.239}^{-0.734}$	$\underline{0.121}^{-0.796}$	$\underline{0.105}^{+0.105}$	$\underline{0.940}^{+0.640}$	$\underline{0.944}^{+0.784}$	$\underline{0.844}^{+0.822}$

5.2 TRAINING

We adopt a two-stage RL-based training framework. Stage I performs supervised fine-tuning to align the model's outputs with reference CoC sequences and task answers, while Stage II uses RL to jointly optimize geometric fidelity, output validity, and answer accuracy.

Supervised fine-tuning. Given an image and query, we train the model to generate the reference CoC and, for reasoning tasks, the final answer. We use teacher forcing and minimize next-token cross-entropy, so each prediction is conditioned on the preceding reference tokens. We fine-tune a separate Qwen3.5-2B model (Qwen Team, 2026) for each task using LoRA (Hu et al., 2022) on both the vision encoder and language model. Coordinates are encoded as integers on Qwen3.5's normalized 0–1000 grid. Direct-answer SFT baselines for reasoning tasks use answer-only targets.

Reinforcement learning. SFT supervises a particular coordinate sequence, although different sequences can trace the same curve. We therefore introduce a *geometric reward* based on continuous Fréchet distance (Section 4.3) to provide feedback on complete generated curves. Each RL example is paired with a reference coordinate sequence and, when applicable, a final answer. The coordinate sequence defines a polyline by joining consecutive coordinates, so computing the reward requires no analytic description of the underlying curve. We combine this geometric reward with standard format and answer rewards (Guo et al., 2025). For sampled response i, the components are: (1) **Geometric reward.** For a nonempty predicted trace $\widehat{P}$ and reference trace P, we use $r_i^{\mathrm{F}} = \exp[-d_{\mathrm{F}}(\Pi(\widehat{P}), \Pi(P))/\tau]$, with distance scale $\tau > 0$. Continuous Fréchet distance compares the full curves in traversal order, using the smallest possible maximum separation between matched points. The reward allows different point counts and spacing; adding points along an existing segment leaves it unchanged. For closed curves, we allow any starting point and either direction; open drawing and writing strokes may also be reversed. For multiple curves or strokes, Hungarian matching pairs predictions with references to maximize total similarity, which we divide by the larger trace count. Queried traces are paired with their references before averaging. Empty and unmatched traces receive zero. (2) **Format reward.** $r_i^{\mathrm{fmt}} = 1$ if the response follows the output format, and 0 otherwise. (3) **Answer reward.** $r_i^{\mathrm{ans}} = 1$ if the final answer agrees with the ground truth, and 0 otherwise. This term is omitted when no separate final answer is required. The total reward is a weighted sum of the three components:

$$R_i = \lambda_{\mathrm{F}} r_i^{\mathrm{F}} + \lambda_{\mathrm{fmt}} r_i^{\mathrm{fmt}} + \lambda_{\mathrm{ans}} r_i^{\mathrm{ans}} \tag{7}$$

where λ_{F}, λ_{fmt}, and λ_{ans} are the non-negative weights. A correctly formatted response that matches the reference curve and any required answer attains the maximum reward, provided it can be represented at the chosen coordinate precision and within the context length limit. For RL, we use

the standard Group Relative Policy Optimization (GRPO; Shao et al., 2024b) initialized from SFT. Detailed training hyperparameters are listed in Appendix E.

5.3 Results

Table 1 shows that CoC SFT improves curve tracing and visual reasoning over the base model and outperforms direct-answer SFT on reasoning tasks. RL further improves performance, matching leading models such as GPT-5.6 Sol and Gemini 3.1 Pro on the selected tasks.

Qualitative analysis. The examples in Appendix G show how CoC improves visual reasoning through explicit curve tracing. However, failure cases remain, with common errors including losing the intended path at crossings, terminating before reaching the target, and inferring incorrect connectivity or containment relations from the traced curves. BabyVision remains particularly challenging because of its longer paths and dense crossings, which make reliable tracing difficult.

CoC prompting. We further examine whether prompting alone can elicit CoC and improve reasoning in models without additional training. We compare direct answering with CoC prompting, which asks VLMs to generate coordinate traces before the final answer (Appendix F). Table 2 shows that GPT models generally benefit from CoC prompting, whereas Gemini models show no consistent gains. A possible explanation is their underlying curve-tracing capability. On tasks that directly evaluate CoC as the output, GPT-family models achieve lower Fréchet distances (Table 1), implying stronger tracing ability. This suggests that CoC is not a model-agnostic prompting strategy. Its effectiveness depends on a model's ability to generate reliable coordinate traces.

Table 2: Accuracy (%) with direct answering (D) and CoC prompting.

Model	BabyVision		Tunnel Vision		CurveBench	
	D	CoC	D	CoC	D	CoC
GPT-5.6 Sol	0.0	0.0	55.2	64.8	80.0	95.6
GPT-5.6 Luna	0.0	5.3	44.8	45.6	53.3	68.9
Gemini 3.8 Flash	5.3	0.0	48.0	45.6	71.1	62.2
Gemini 3.1 Pro	5.3	0.0	41.6	46.4	66.7	66.7
Claude Opus 5	5.3	0.0	33.6	36.8	44.4	51.1
Qwen3.5 2B	0.0	0.0	16.0	4.8	2.2	0.0

When CoC helps. Direct-answer SFT performs comparably on TRACER-Bench, achieving 84% accuracy compared with 92% for CoC SFT (Table 1). One explanation is that distinct path colors can reveal endpoint correspondences directly without tracing (Figure 3). CoC is therefore most useful when identifying a connection requires explicitly following the path, consistent with Theorem 12. To test this, we compare CoC with direct answering on the Unique Colors variant of Circuit Connections, where each wire has a distinct color. CoC's accuracy advantage compared to direct answering decreases from 18.4 percentage points in the Single-Color setting to 2.4 points with Unique Colors, supporting the effectiveness of CoC when direct visual shortcuts are unavailable.

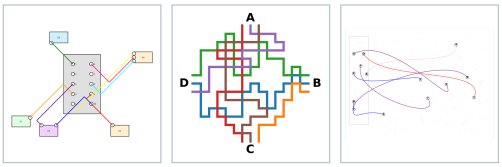


Figure 3: Distinct colors can reveal connections without tracing. Left: Tunnel Vision Circuit Connections Unique Colors; middle: BlindTest; right: TRACER-Bench, where some paths share colors.

6 Conclusion

We introduced Chain-of-Coordinates (CoC), which enables VLMs to represent curves and maintain spatial state through coordinate sequences generated as text. Our analysis establishes representability, learnability, error bounds, and a computational advantage over direct answering on a family of connectivity problems. We combine this formulation with an RL-based training framework using a Fréchet-based geometric reward. Experiments across six tasks demonstrate improved tracing and visual reasoning. We also examine when CoC helps: it is more effective when no visual shortcuts directly reveal connectivity, and prompting benefits depend on the model's ability to follow curves reliably. Our work provides a theoretical and empirical foundation for using explicit coordinate sequences to connect visual perception with sequential reasoning.

REPRODUCIBILITY STATEMENT

Section 5 describes data preparation and training, with hyperparameters in Appendix E and CoC prompting instructions in Appendix F.

AI USE STATEMENT

Our synthetic data generation code and experimental code were written by or with assistance from generative AI. Generative AI also assisted with polishing the manuscript.

REFERENCES

David A. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. *Journal of Computer and System Sciences*, 38(1):150–164, 1989. doi: 10.1016/0022-0000(89)90037-8. URL https://doi.org/10.1016/0022-0000(89)90037-8.

Shmuel Berman and Jia Deng. VLMs have tunnel vision: Evaluating nonlocal visual reasoning in leading VLMs. In *Advances in Neural Information Processing Systems*, volume 38, pp. 78972–78993, 2025. doi: 10.52202/085713-2642. URL https://arxiv.org/abs/2507.13361.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. URL https://web.stanford.edu/~boyd/cvxbook/.

Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3–4):231–357, 2015. doi: 10.1561/2200000050. URL https://arxiv.org/abs/1405.4980.

Liang Chen, Weichu Xie, Yiyan Liang, Hongfeng He, Hans Zhao, Zhibo Yang, Zhiqi Huang, Haoning Wu, Haoyu Lu, Y. charles, Yiping Bao, Yuantao Fan, Guopeng Li, Haiyang Shen, Xuanzhong Chen, Wendong Xu, Shuzheng Si, Zefan Cai, Wenhao Chai, Ziqi Huang, Fangfu Liu, Tianyu Liu, Baobao Chang, Ming Wu, Xiaobo Hu, Kaiyuan Chen, Yixin Ren, Yang Liu, Yuan Gong, and Kuan Li. BabyVision: Visual reasoning beyond language. arXiv preprint arXiv:2601.06521, 2026. URL https://arxiv.org/abs/2601.06521.

David Chiang. Transformers in uniform TC^0. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=ZA7D4nQuQF.

Brandon Collins, Logan Bolton, Hung Huy Nguyen, Mohammad Reza Taesiri, Trung Bui, and Anh Totti Nguyen. SketchVLM: Vision language models can annotate images to explain thoughts and guide users. arXiv preprint arXiv:2604.22875, 2026. URL https://arxiv.org/abs/2604.22875.

Changfei Fu, Guangcheng Chen, Aoxiang Gu, Haoxiang Liang, Wenjun Xu, and Hong Zhang. TrajVLN: Learning pixel-space interaction via autoregressive trajectory generation. arXiv preprint arXiv:2607.10744, 2026. URL https://arxiv.org/abs/2607.10744.

Google Creative Lab. The Quick, Draw! dataset. Dataset and documentation, 2017. URL https://github.com/googlecreativelab/quickdraw-dataset.

Daya Guo, Dejian Yang, Haowei Zhang, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z. URL https://doi.org/10.1038/s41586-025-09422-z.

Tengda Guo, Jie Leng, Hanlei Li, Yaoyuan Liang, Qingyue Zhang, Dian Yang, Mingyu Zhang, Yuhua Fu, and Shao-Lun Huang. CAVE: A structured credit assignment approach for fragmented visual evidence reasoning. arXiv preprint arXiv:2605.16416, 2026. URL https://arxiv.org/abs/2605.16416.

Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002. URL https://pi.math.cornell.edu/~hatcher/AT/AT.pdf.

Hyesoo Hong, Minsoo Kim, Wonje Jeung, Sangyeon Yoon, Dongjae Jeon, and Albert No. VLMs trace without tracking: Diagnosing failures in visual path following. arXiv preprint arXiv:2605.15672, 2026. URL https://arxiv.org/abs/2605.15672.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-4423. URL https://arxiv.org/abs/2406.09403.

Chaoya Jiang, Yongrui Heng, Wei Ye, Haiyang Xu, Ming Yan, Ji Zhang, Fei Huang, and Shikun Zhang. VLM-R^3: Region recognition, reasoning, and refinement for enhanced multimodal chain-of-thought. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2143. URL https://arxiv.org/abs/2505.16192.

Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. doi: 10.1126/science.aab3050. URL https://doi.org/10.1126/science.aab3050.

Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. Imagine while reasoning in space: Multimodal visualization-of-thought. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 36340–36364. PMLR, 2025. URL https://proceedings.mlr.press/v267/li25cz.html.

Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of thought empowers transformers to solve inherently serial problems. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3EWTEy9MTM.

Tiedong Liu and Wee Sun Lee. Can vision-language models solve the shell game? arXiv preprint arXiv:2603.08436, 2026. URL https://arxiv.org/abs/2603.08436.

Ruijie Lu, Yiyang Ma, Xiaokang Chen, Lingxiao Luo, Zhiyu Wu, Zizheng Pan, Xingchao Liu, Yutong Lin, Hao Li, Wen Liu, Zhewen Hao, Xi Gao, Shaoheng Nie, Yixuan Wei, Zhenda Xie, Ting Chen, and Gang Zeng. Thinking with visual primitives, 2026. URL https://fengweifeng.com/deepseek-papers/pdfs/Thinking_with_Visual_Primitives.pdf. Technical report.

Sachit Menon, Richard Zemel, and Carl Vondrick. Whiteboard-of-thought: Thinking step-by-step across modalities. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 20016–20031. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.1117. URL https://aclanthology.org/2024.emnlp-main.1117/.

William Merrill and Ashish Sabharwal. A logic for expressing log-precision transformers. In *Advances in Neural Information Processing Systems*, volume 36, pp. 52453–52463, 2023a. doi: 10.52202/075280-2284. URL https://arxiv.org/abs/2210.02671.

William Merrill and Ashish Sabharwal. The parallelism tradeoff: Limitations of log-precision transformers. *Transactions of the Association for Computational Linguistics*, 11:531–545, 2023b. doi: 10.1162/tacl_a_00562. URL https://aclanthology.org/2023.tacl-1.31/.

William Merrill and Ashish Sabharwal. The expressive power of transformers with chain of thought. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NjNGlPh8Wh.

Amirreza Mohseni, Mona Mohammadi, Morteza Saghafian, and Naser Talebizadeh Sardari. CurveBench: A benchmark for exact topological reasoning over nested Jordan curves. arXiv preprint arXiv:2605.14068, 2026. URL https://arxiv.org/abs/2605.14068.

Artemis Panagopoulou, Aveek Purohit, Achin Kulshrestha, Soroosh Yazdani, and Mohit Goyal. MapTrace: Scalable data generation for route tracing on maps. arXiv preprint arXiv:2512.19609, 2025. URL https://arxiv.org/abs/2512.19609.

Clara Petrova, Zhuo Chen, and Marin Soljačić. TraversalBench: Challenging paths to follow for vision language models. arXiv preprint arXiv:2604.10999, 2026. URL https://arxiv.org/abs/2604.10999.

Qwen Team. Qwen3.5-2B. Model card, 2026. URL https://huggingface.co/Qwen/Qwen3.5-2B.

Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. Vision language models are blind. In *Proceedings of the Asian Conference on Computer Vision*, pp. 18–34, 2024. URL https://arxiv.org/abs/2407.06581v1.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual CoT: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 8612–8642, 2024a. doi: 10.52202/079017-0275. URL https://arxiv.org/abs/2403.16999.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300, 2024b. URL https://arxiv.org/abs/2402.03300.

Tianhui Song, Haoyu Lu, Hao Yang, Lin Sui, Haoning Wu, Zaida Zhou, Zhiqi Huang, Yiping Bao, Y.Charles, Xinyu Zhou, and Limin Wang. Towards pixel-level VLM perception via simple points prediction. arXiv preprint arXiv:2601.19228, 2026. URL https://arxiv.org/abs/2601.19228.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19(70): 1–57, 2018. URL https://jmlr.org/papers/v19/18-188.html.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022. doi: 10.52202/068431-1800. URL https://proceedings.neurips.cc/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Junfei Wu, Jian Guan, Kaituo Feng, Qiang Liu, Shu Wu, Liang Wang, Wei Wu, and Tieniu Tan. Reinforcing spatial reasoning in vision-language models with interwoven thinking and visual drawing. In *Advances in Neural Information Processing Systems*, volume 38, pp. 143297–143330, 2025. doi: 10.52202/085713-4797. URL https://arxiv.org/abs/2506.09965.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13084–13094, 2024. URL https://arxiv.org/abs/2312.14135.

Fan Yang, Shurong Zheng, Hongyin Zhao, Yufei Zhan, Xin Li, Yousong Zhu, Chaoyang Zhao, Ming Tang, and Jinqiao Wang. TraceVision: Trajectory-aware vision-language model for human-like spatial understanding. arXiv preprint arXiv:2602.19768, 2026a. URL https://arxiv.org/abs/2602.19768.

Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. Machine mental imagery: Empower multimodal reasoning with latent visual tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33510–33520, 2026b. URL https://arxiv.org/abs/2506.17218.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. DeepEyes: Incentivizing "thinking with images" via reinforcement learning. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=xUyMXkI958.

A Discrete Implementation

A.1 Coordinate encoding

For the proofs, we store rounded coordinates as integers and recover their locations using a fixed scale. Choose an integer $B \geq 1$, giving coordinate spacing $1/B$. With $B = 1000$, we store $(0.123, 0.456)$ as $(123, 456)$; dividing by 1000 recovers the point. Generally, $p = (u/B, v/B)$ with $u, v \in \{0, \ldots, B\}$. Points between grid locations are rounded by Q_B from Section 4.3. Rounding may move a point slightly; writing the rounded numbers as text adds no further error.

Models write text in pieces called tokens. GPT-5 splits consecutive digits from the left into groups of up to three; Qwen3.5 splits them individually. For `123456`, this gives `123 | 456` versus `1 | 2 | 3 | 4 | 5 | 6`. The bars show token boundaries, not output characters. The value is unchanged, but the models need different numbers of token predictions to write it.

For the proofs, each coordinate has the same number of token positions, making earlier points easy to locate. With $B = 1000$, we need room for 1000: the single-digit format writes 123 as `0123`, while the three-digit format writes `000 | 123`. Leading zeros preserve the value. Fixed separators distinguish coordinates, successive points, and curves.

To state the token count, let $b \geq 2$ be a fixed integer number base (radix), with one token per base-b digit: $b = 10$ for individual decimal digits, or $b = 1000$ for complete three-digit groups. Each coordinate uses

$$\ell_{B,b} = \lceil \log_b(B + 1) \rceil$$

number tokens, including leading zeros; $B + 1$ counts the values from 0 through B. A point (a coordinate record) takes $2\ell_{B,b} + O(1)$ tokens, including a fixed number of separators. This is $O(\log(B + 1))$ for fixed b, as used in Section 4.4. Precision may increase with input size; the number base and available tokens remain fixed.

A.2 Decoder primitives and local state access

For the transformer family in Section 4.1, write $M(\bar{p}) = \mathbf{1}[\bar{p} \in M]$, with zero occupancy outside Λ_n. Set $B_{\mathrm{idx}} = \max\{1, n - 1\}$, so pixel indices and normalized coordinates satisfy $p_t = \bar{p}_t / B_{\mathrm{idx}}$ and a path with T steps has $K = T+1$ points. For $n = 1$, the grid contains only $(0, 0)$ and $B_{\mathrm{idx}} = 1$. At the start, use a marker $\bar{p}_{-1} = \perp$ outside the grid; the factor $\mathbf{1}[t > 0]$ makes $b_0 = 0$.

Let $N_{\mathrm{pix}} = n^2$, let the patch side s_{patch} divide n, and let $L_{\mathrm{ctx}} \geq 1$ bound context length in tokens. Rectangular rasters use separate axis ranges. Computation uses exact arithmetic without mandatory layer normalization or lossy bottlenecks. Width, heads, positional tables, and context may grow polynomially with n; for the simple-path construction, context must hold the query, up to n^2 path records, and termination.

Greedy decoding over a fixed finite vocabulary $\mathcal{V}$ uses a fixed tie rule:

$$z_\ell = \arg\max_{v \in \mathcal{V}} p_\theta(v \mid I, q, z_{<\ell}). \tag{8}$$

Here p_θ is the model's next-token distribution, and $z_{<\ell}$ contains the tokens already generated. Generation begins with no output tokens and uses the coordinate format in Appendix A.1. The function F_θ in Equation (1) predicts one complete point, which may take several token predictions. At $t = 0$, the list of earlier coordinates in that equation is empty. Separators, stopping tokens, and answer tokens are not points in P.

Lemma A.1 (Boolean decoder primitives)**.** *For any AC^0 function of the image pixels and a fixed number of addressed prefix records, the transformer family of Section 4.1 can be constructed to compute it and write a selected grid location as a coordinate record.*

Proof. Pixel access. The patch embedding retains the s^2_{patch} occupancy bits and a disjoint one-hot patch-position vector. For local occupancy μ_a and position channel e_j, the gate $\mathrm{ReLU}(\mu_a + e_j - 1)$ equals one exactly at an occupied pixel in patch j. These N_{pix} channels assign each global pixel to exactly one patch token. An attention head masked to the $N_{\mathrm{patch}} = N_{\mathrm{pix}}/s^2_{\mathrm{patch}}$ visual patch tokens

assigns them equal logits. It therefore averages their gated vectors, and multiplication by N_{patch} recovers the full mask. The required depth is constant and the width is polynomial in n.

Prefix access. Each radix symbol is represented internally by a fixed-length binary code. Since the radix is fixed, symbol equality is a constant-size Boolean test. The fixed vocabulary's delimiters and termination tokens also have fixed-length binary codes. Assign one-hot keys to the L_{ctx} prefix positions. A positional query selects the required completed record, with offset determined by the output phase. Let L_{att} count all positions visible to this head, including visual or query positions unless they are masked out. Set the selected logit to κ_{att} and all other logits to zero, with $e^{\kappa_{\text{att}}} > 3(L_{\text{att}} - 1)$. The selected weight $e^{\kappa_{\text{att}}}/(e^{\kappa_{\text{att}}} + L_{\text{att}} - 1)$ is then greater than $3/4$. Binary value channels consequently have error less than $1/4$. The ReLU map

$$\text{round}_{\text{bit}}(z) = 2\,\text{ReLU}(z - 1/4) - 2\,\text{ReLU}(z - 3/4) \tag{9}$$

recovers each bit exactly: it is zero for $z \le 1/4$ and one for $z \ge 3/4$. Separate heads address each required completed record. The marked-path construction uses the last two records, while the endpoint matching construction addresses its preceding boundary record. Polynomial-size positional tables specify the offsets, including phases within an unfinished record; the query supplies the initial sentinels.

Boolean transitions. For binary inputs $\zeta_i, \zeta \in \{0, 1\}$, conjunction of k inputs, negation, and disjunction are represented by $\text{ReLU}(\sum_i \zeta_i - k + 1)$, $1 - \zeta$, and $\text{ReLU}(\sum_i \zeta_i) - \text{ReLU}(\sum_i \zeta_i - 1)$, respectively. Replacing each gate of a constant-depth polynomial-size Boolean circuit therefore gives a constant-depth polynomial-size ReLU circuit.

Coordinate serialization. For the fixed radix b, the integer-grid field width is $\ell_{\text{idx}} = \ell_{B_{\text{idx}},b} = \max\{1, \lceil \log_b n \rceil\}$. Each field has ℓ_{idx} padded radix symbols, including one zero symbol when $n = 1$. A completed coordinate field is decoded by comparing its padded symbol sequence with the possible grid-value encodings; conversely, the selected grid value and output phase determine the next radix symbol by lookup. For each axis, ℓ_{idx} heads recover the symbol codes, and conjunctions of symbol equality tests compare the field with all n possible values. Conjunction of the two resulting one-hot axis vectors yields a one-hot location among n^2 possibilities. Each output symbol is recomputed from the same completed prefix records. The number of heads and gates is polynomial, and the depth is constant. Delimiters and `EOS` occupy fixed output phases. Expose the one-hot indicator of the selected next token as the final decoder feature. The identity token readout then sets the correct-token logit to one and all remaining logits to zero, giving exact greedy decoding. □

Proof of Lemma 1. Use Lemma A.1 to expose the occupancy bits and one-hot locations of the last two completed records, with a predecessor sentinel at the first record. For each fixed $d \in \mathcal{D}$,

$$m_t(d) = \sum_{\bar{p} \in \Lambda_n} \mathbf{1}[\bar{p}_t = \bar{p}] M(\bar{p} + d), \qquad b_t(d) = \mathbf{1}[t > 0] \sum_{\bar{p} \in \Lambda_n} \mathbf{1}[\bar{p}_t = \bar{p}] \mathbf{1}[\bar{p}_{t-1} = \bar{p} + d]. \tag{10}$$

Occupancy outside the grid is zero. Every product is a binary conjunction and at most one summand is nonzero. Equality and the sentinel convention therefore implement these features in constant depth, with $b_0 = 0$. For an in-grid displacement, the location–displacement pair indexes the next coordinate and its padded radix symbols. An out-of-grid displacement receives `EOS` as a default; it is never chosen on a correct admissible prefix. All radix symbols are recomputed from the unchanged completed records, so the same chosen displacement is used throughout the next record.

For any fixed U, greedy selection with the fixed tie rule gives a finite Boolean prediction table, implemented by a constant-size circuit for fixed $\mathcal{D}$. This covers both the explicit scores and a learned readout. Theorem 3 optimizes the linear head before this deployment compilation. □

Finite-memory extension. For Proposition 5, encode the finite pixel alphabet by a fixed number of binary channels and retain these channels in the patch embeddings. The pixel-access construction applies channelwise. Addressed prefix access recovers the fixed number of recent coordinate records, with sentinels for missing initial records. These coordinates select the local image patch, and a lookup implements the fixed local-state computation and its one-hot output. Because the patch size, pixel alphabet, and number of accessed records are fixed, the lookup has constant depth and polynomial size in the grid size. The learned finite prediction table is implemented as above, and every output symbol is recomputed from the same completed records until the next coordinate is

complete. Traversals may revisit coordinates, so their record budget may exceed n^2 while remaining polynomial in n.

A.3 Enumeration and paths with supplied termination

Proposition A.2 (General discrete representability)**.** *The transformers of Lemma A.1 can (i) list all foreground pixels of a binary mask in raster order, then* `EOS`*; and (ii) given the start and step count T (or terminal index), recover a simple marked path whose successor at each nonterminal point is its unique marked neighbor within a fixed distance h_{loc}, excluding the predecessor.*

Proof. Enumeration. For an integer index $(u, v) \in \Lambda_n$, let $\mathrm{rank}(u, v) = nv + u$. Write M_j for the occupancy of rank j, where $0 \le j < N_{\mathrm{pix}}$. Let j_{prev} be the last emitted rank, initially -1. Prefix access gives $\chi_j = \mathrm{ReLU}(M_j + \mathbf{1}[j > j_{\mathrm{prev}}] - 1)$, the eligible mask. Define

$$c_j = \mathrm{ReLU}\left(\chi_j - \sum_{k<j} \chi_k\right), \qquad c_{\texttt{EOS}} = \mathrm{ReLU}\left(1 - \sum_j \chi_j\right). \tag{11}$$

If any eligible pixel exists, $c_j = 1$ precisely at its smallest rank and $c_{\texttt{EOS}} = 0$; otherwise all $c_j = 0$ and $c_{\texttt{EOS}} = 1$. Each transition has constant depth and polynomial size. The emitted ranks strictly increase, enumerating every occupied pixel exactly once in at most N_{pix} records. An empty mask emits `EOS` immediately.

Marked paths. Let $\bar{P} = (\bar{p}_0, \ldots, \bar{p}_T)$ be the simple marked path starting at the point $\bar{p}_0$ supplied in the query. At the first step, eligible successors are points of $M \setminus \{\bar{p}_0\}$ within distance h_{loc} of $\bar{p}_0$. At step $0 < t < T$, they are points of $M \setminus \{\bar{p}_{t-1}, \bar{p}_t\}$ within distance h_{loc} of $\bar{p}_t$. One-hot prefix indices address the tables for $\bar{p}, \bar{q} \in \Lambda_n$

$$A_{\bar{p},\bar{q}} = \mathbf{1}[\|\bar{p} - \bar{q}\|_2 \le h_{\mathrm{loc}}] \quad \text{and} \quad \mathbf{1}[\bar{p} = \bar{q}]. \tag{12}$$

Conjunction with occupancy and the exclusion indicators yields a one-hot candidate vector by the unique-successor assumption. The coordinate readout emits that candidate. Induction from the supplied $\bar{p}_0$ gives the full integer sequence $\bar{P}$. Dividing each point by B_{idx} gives the normalized CoC P. At the terminal point, equality with the supplied step count or terminal index selects `EOS` before successor evaluation. For $T = 0$, the output is $\bar{p}_0$ followed by `EOS`. Since $\bar{P}$ is simple, its counters are bounded by N_{pix}; the lookup tables and transition circuits have polynomial size and constant depth. □

B Learning the Local Rule

B.1 Training setup and proof of Theorem 3

Write $q_{\mathcal{D}} = |\mathcal{D}|$, so the features in Equation (5) have dimension $2q_{\mathcal{D}} + 1$. Let $\mathcal{S}_{\mathcal{D}}$ contain every feature–label pair (h_t, a_t) occurring on the paths of Theorem 2, over all grid sizes. These pairs associate the local feature vectors in Section 4.2 with their target labels: the features specify the occupied neighbors and previous point, and the label specifies the next direction or `EOS`. Equation (2) makes the label unique for each feature vector.

Use a finite, nonempty training collection $\{(h_i, a_i)\}_{i=1}^{N_{\mathrm{tr}}}$ from $\mathcal{S}_{\mathcal{D}}$, allowing repeated examples. Write $u_a^\top$ for row a of U and minimize summed softmax cross-entropy:

$$\mathcal{L}(U) = \sum_{i=1}^{N_{\mathrm{tr}}} \left[\log \sum_{a \in \mathcal{A}} \exp(u_a^\top h_i) - u_{a_i}^\top h_i\right]. \tag{13}$$

Only U is trained. Gradient descent starts from any U_0 and uses a fixed learning rate satisfying

$$0 < \alpha \le 1/\beta, \qquad \beta = \sum_{i=1}^{N_{\mathrm{tr}}} \|h_i\|_2^2. \tag{14}$$

At generation time, the highest-scoring class specifies the next direction or `EOS`, and the coordinate writer emits the corresponding record or stopping token. The same U is used across grid sizes.

Lemma B.1 (Softmax regression on separable features)**.** *Suppose $U^\star$ separates labeled vectors (h_i, a_i) with margin $\gamma > 0$. From any initialization, gradient descent on Equation* (13) *with $0 < \alpha \le 1/\beta$, $\beta = \sum_i \|h_i\|_2^2 > 0$, drives $\mathcal{L}$ to zero; once $\mathcal{L} < \log 2$, every example is classified correctly.*

Proof. The margin condition is $(U^\star h_i)_{a_i} \ge (U^\star h_i)_a + \gamma$ for all i and $a \ne a_i$. Write $\ell_i(U) = \log \sum_v \exp(u_v^\top h_i) - u_{a_i}^\top h_i$. Each ℓ_i is convex (Boyd & Vandenberghe, 2004, Section 3.1.5). In a matrix direction A, its Hessian quadratic form is the variance of $(Ah_i)_v$ under the softmax distribution and is bounded by $\|Ah_i\|_2^2 \le \|A\|_F^2 \|h_i\|_2^2$. Summing over i gives β-smoothness. For $\tau > 0$, the separation assumption gives

$$\ell_i(\tau U^\star) \le \log\bigl(1 + (|\mathcal{A}| - 1)e^{-\tau\gamma}\bigr). \tag{15}$$

The right-hand side tends to zero, hence $\inf_U \mathcal{L}(U) = 0$.

For $U_{k+1} = U_k - \alpha \nabla \mathcal{L}(U_k)$ and $\alpha \le 1/\beta$, smoothness yields $\mathcal{L}(U_{k+1}) \le \mathcal{L}(U_k) - \alpha \|\nabla \mathcal{L}(U_k)\|_F^2/2$ (Bubeck, 2015, Section 3.2). Convexity and expansion of the squared distance to any comparator U_{ref} imply

$$2\alpha\bigl(\mathcal{L}(U_{k+1}) - \mathcal{L}(U_{\mathrm{ref}})\bigr) \le \|U_k - U_{\mathrm{ref}}\|_F^2 - \|U_{k+1} - U_{\mathrm{ref}}\|_F^2. \tag{16}$$

Telescoping and monotonicity of $\mathcal{L}(U_k)$ give

$$\mathcal{L}(U_k) \le \mathcal{L}(U_{\mathrm{ref}}) + \frac{\|U_0 - U_{\mathrm{ref}}\|_F^2}{2\alpha k}. \tag{17}$$

With $U_{\mathrm{ref}} = \tau U^\star$, taking $k \to \infty$ and then $\tau \to \infty$ gives $\mathcal{L}(U_k) \to 0$. This loss limit does not require a minimizer (Soudry et al., 2018, Section 4.1).

If $\mathcal{L}(U) < \log 2$, nonnegativity of the example losses implies $\ell_i(U) < \log 2$ for every i. Each target class then has probability greater than $1/2$ and is the unique greedy maximizer. □

Proof of Theorem 3.

Proof. Equation (4) is linear in h_t and supplies a margin-one separator by Theorem 2. The bias coordinate gives $\beta \ge N_{\mathrm{tr}} > 0$, so Lemma B.1 yields $\mathcal{L}(U_k) \to 0$ under Equation (14). Eventually $\mathcal{L}(U_k) < \log 2$, so every covered state's correct class has probability greater than $1/2$ and is the unique greedy maximizer. Complete coverage gives correctness on every state in $\mathcal{S}_\mathcal{D}$. For any new path satisfying Equation (2), the initial state and each state reached after a correct sequence of points belong to this set. Induction therefore recovers the path. □

B.2 State coverage and tied parameters

The singleton, start, interior, and terminal cases of Equation (2) give the size-independent bound

$$|\mathcal{S}_\mathcal{D}| \le 1 + 2q_\mathcal{D} + q_\mathcal{D}(q_\mathcal{D} - 1) = q_\mathcal{D}^2 + q_\mathcal{D} + 1. \tag{18}$$

Every realizable state is a singleton, a start with direction d, a terminal with predecessor direction d, or an interior with distinct predecessor and successor directions $d_-, d_+ \in \mathcal{D}$. At an interior, $d_- - d_+ \notin \mathcal{D}$: otherwise the predecessor and successor are adjacent, contradicting Equation (2) at the predecessor. Conversely, any such ordered pair is realized by the three-point path $(d_-, 0, d_+)$, translated into a sufficiently large grid. Singleton and one-edge paths realize the other types. Hence

$$|\mathcal{S}_\mathcal{D}| = 1 + 2q_\mathcal{D} + \bigl|\{(d_-, d_+) \in \mathcal{D}^2 : d_- \ne d_+,\ d_- - d_+ \notin \mathcal{D}\}\bigr|. \tag{19}$$

For the eight-neighbor set, 32 ordered interior pairs are admissible, giving $1 + 16 + 32 = 49$ states. The other 24 of the 73 formal patterns violate Equation (2). A collection of paths of at most three points suffices to cover every realizable state.

For a fixed local state $h = h_t$, write $m_d = m_t(d)$ and $b_d = b_t(d)$. An optional tied head uses $s_d = \theta_m m_d + \theta_b b_d$ and $s_{\texttt{EOS}} = c_0 + c_1 \sum_d m_d + c_2 \sum_d b_d$. The parameter choice $\vartheta = (\theta_m, \theta_b, c_0, c_1, c_2) = (2, -2, 1, 0, 0)$ gives the same separator. Direction symmetry allows training on one representative of each of the four state types. Writing the logits as $A(h)\vartheta$, a direction row of $A(h)$ is $(m_d, b_d, 0, 0, 0)$;

the EOS row is $(0, 0, 1, \sum m_d, \sum b_d)$. The smoothness argument for Lemma B.1 applies with the sufficient parameter-space bound

$$\beta_{\text{tied}} = \sum_i \|A(h_i)\|_F^2. \tag{20}$$

The explicit parameter vector separates all states. From any initialization, gradient descent with fixed step $0 < \alpha \le 1/\beta_{\text{tied}}$ therefore drives the summed representative loss below $\log 2$ in finitely many iterations. Every other state of an occurring type is a direction permutation of its representative, and the tied architecture permutes direction logits while leaving EOS unchanged. The unique correct maximizer on each representative thus gives correctness on that entire type.

B.3 Robustness of a learned readout

Write $\widetilde{m}_t, \widetilde{b}_t$ for approximate local features. The error assumption in Corollary 4 is

$$\|\widetilde{m}_t - m_t\|_\infty \le \varepsilon, \qquad \|\widetilde{b}_t - b_t\|_\infty \le \varepsilon, \tag{21}$$

at every correct path prefix. The explicit scores tolerate $\varepsilon < 1/4$; a learned predictor instead has a bound determined by its score differences and weights.

Let U be an iterate from Theorem 3 that classifies every state in $\mathcal{S}_\mathcal{D}$ with a unique maximizer. Write $u_a = (w_a, c_a)$, where $w_a \in \mathbb{R}^{2q_\mathcal{D}}$ contains the feature weights and c_a is the bias. Define

$$\gamma_U = \min_{(h,a_\star)\in\mathcal{S}_\mathcal{D}} \min_{a\neq a_\star} (u_{a_\star} - u_a)^\top h > 0, \qquad L_U = \max_{a\neq a'} \|w_a - w_{a'}\|_1. \tag{22}$$

The margin is positive because the state set is finite. For a state $(h, a_\star)$, a competing class $a \neq a_\star$, and $\widetilde{h} = (\widetilde{m}, \widetilde{b}, 1)$ satisfying Equation (21), the bias coordinate is unchanged and

$$(u_{a_\star} - u_a)^\top \widetilde{h} \ge \gamma_U - \varepsilon L_U > 0 \qquad \text{whenever} \qquad \varepsilon L_U < \gamma_U. \tag{23}$$

Hence the recovery conclusion of Corollary 4 holds for the learned head whenever $\varepsilon L_U < \gamma_U$.

B.4 Local rules using recent points

Consider grid traversals $\bar{P} = (\bar{p}_0, \ldots, \bar{p}_T)$ with moves in $\mathcal{D}$. At each reference prefix, a fixed procedure computes z_t from a fixed-size local patch over a finite pixel alphabet and a bounded number of recent coordinate records. Missing initial records use sentinel markers. Let $\mathcal{S}$ be the finite set of states occurring over the whole family. A rule $f : \mathcal{S} \to \mathcal{D} \cup \{\texttt{EOS}\}$ satisfies

$$f(z_t) = \begin{cases} \bar{p}_{t+1} - \bar{p}_t, & t < T, \\ \texttt{EOS}, & t = T. \end{cases}$$

Represent z_t by $h_t = e_{z_t}$, a vector with a 1 at the position for that state and 0 elsewhere. Train a linear predictor on $N_{\text{tr}} \ge 1$ examples $(e_z, f(z))$ covering $\mathcal{S}$, allowing repeats. Use the summed loss in Equation (13), any initialization, and gradient descent with $0 < \alpha \le 1/N_{\text{tr}}$. Appendix A implements these features and the resulting predictor.

Proof of Proposition 5.

Proof. The matrix $U^\star_{a,z} = \mathbf{1}[a = f(z)]$ separates all state–label pairs with margin one. Since each feature has unit squared norm, Lemma B.1 applies with $\beta = N_{\text{tr}}$, and the summed loss eventually falls below $\log 2$. Complete state coverage then gives the correct unique greedy prediction on every state in $\mathcal{S}$. Induction from the supplied start recovers each next coordinate and finally EOS. □

C Continuous Representation

C.1 Distance and coordinate errors

For continuous curves $\gamma, \xi : [0, 1] \to \Omega$, continuous Fréchet distance is

$$d_{\text{F}}(\gamma, \xi) = \inf_{\alpha,\beta} \max_{u\in[0,1]} \|\gamma(\alpha(u)) - \xi(\beta(u))\|_2, \tag{24}$$

where $\alpha, \beta : [0,1] \to [0,1]$ are continuous nondecreasing surjections. They preserve traversal order while allowing different speeds and pauses. Curves on other compact intervals are rescaled to $[0,1]$; curves at zero distance are identified. For a nonempty point list P, $\Pi(P)$ joins consecutive points with straight segments; a singleton gives a constant curve. Distances between point lists mean distances between these connected curves.

For an integer $B \geq 1$, the coordinate grid is $\mathcal{G}_B = \{0, 1/B, \ldots, 1\}^2 = \{\bar{p}/B : \bar{p} \in \Lambda_{B+1}\}$. The map $Q_B : \Omega \to \mathcal{G}_B$ rounds each coordinate to its nearest grid value, with a fixed rule for ties. Each coordinate moves by at most $1/(2B)$, so $\|Q_B(p) - p\|_2 \leq \eta_B = \sqrt{2}/(2B)$.

For equally long CoCs P, Q, the calculation behind Lemma 6 is, for $u \in [0,1]$,

$$\|(1-u)(p_i - q_i) + u(p_{i+1} - q_{i+1})\|_2 \leq (1-u)\|p_i - q_i\|_2 + u\|p_{i+1} - q_{i+1}\|_2 \leq \delta, \tag{25}$$

provided $\max_i \|p_i - q_i\|_2 \leq \delta$. Matching corresponding segments gives $d_{\mathrm{F}}(\Pi(P), \Pi(Q)) \leq \delta$; a singleton satisfies the same bound directly.

C.2 Curve approximation

For Proposition 7, unit speed gives $\|\gamma''(s)\|_2 \leq \kappa$. Write the sampling times as $0 = s_0 < \cdots < s_J = L$, with $s_{i+1} - s_i \leq h$. The samples $p_i = \gamma(s_i)$ and predictions $\widehat{p}_i$ form CoCs P and $\widehat{P}$ of length $J+1$.

Proof. Let $P = (p_0, \ldots, p_J)$ with $p_i = \gamma(s_i)$. On a sampling interval $[s_-, s_+]$, let γ_{lin} be the affine chord interpolant and set $E = \gamma - \gamma_{\mathrm{lin}}$. For any unit vector v, define

$$\psi_v(s) = v^\top E(s) - \frac{\kappa}{2}(s - s_-)(s_+ - s). \tag{26}$$

The curvature bound gives $\psi_v''(s) = v^\top \gamma''(s) + \kappa \geq 0$, so ψ_v is convex. Its endpoint values are zero, hence $\psi_v(s) \leq 0$ on $[s_-, s_+]$. Taking the supremum over unit vectors yields

$$\|E(s)\|_2 \leq \frac{\kappa}{2}(s - s_-)(s_+ - s) \leq \frac{\kappa(s_+ - s_-)^2}{8}. \tag{27}$$

Matching each sampling interval to its chord and concatenating these matches preserves traversal order, giving $d_{\mathrm{F}}(\gamma, \Pi(P)) \leq \kappa h^2/8$. Furthermore,

$$\begin{aligned} \max_i \|\widehat{p}_i - p_i\|_2 &\leq \max_i \|\widehat{p}_i - Q_B(p_i)\|_2 + \max_i \|Q_B(p_i) - p_i\|_2 \\ &\leq e + \eta_B. \end{aligned} \tag{28}$$

Lemma 6 therefore gives $d_{\mathrm{F}}(\Pi(P), \Pi(\widehat{P})) \leq e + \eta_B$. The triangle inequality proves Equation (6). □

Continuous Fréchet distance is unchanged by subdividing existing segments without changing the oriented polygonal curve; discrete Fréchet distance can depend on the selected vertices.

C.3 Masks from closed boundary traces

For Corollary 8, assume the sampling conditions above, $\gamma(0) = \gamma(L)$, and that γ is injective on $[0, L)$. Write $\Gamma = \gamma([0, L])$ and let R be the bounded component of $\mathbb{R}^2 \setminus \Gamma$. Require $\widehat{p}_J = \widehat{p}_0$ and let $\widehat{R} \subseteq \Omega$ be the even–odd fill of $\Pi(\widehat{P})$. Even–odd filling assigns foreground to an off-contour point when the contour has odd winding number about it; points on the contour may be assigned by any fixed convention. The predicted polygon need not be simple. With $\delta = \kappa h^2/8 + \eta_B + e$, the claim is

$$R \triangle \widehat{R} \subseteq \big\{x \in \Omega : \operatorname{dist}(x, \Gamma) \leq \delta\big\},$$

where $\triangle$ denotes the set of points assigned different labels by the two masks.

Proof. Parameterize the predicted polygonal curve by linearly interpolating its vertices at the reference sampling times:

$$\xi(s) = (1-u)\widehat{p}_i + u\widehat{p}_{i+1}, \qquad u = \frac{s - s_i}{s_{i+1} - s_i}, \quad s \in [s_i, s_{i+1}].$$

The chord-error estimate in Equation (27) and the vertex bound $\|\widehat{p}_i - p_i\|_2 \le \eta_B + e$ give

$$\sup_{s\in[0,L]} \|\xi(s) - \gamma(s)\|_2 \le \frac{\kappa h^2}{8} + \eta_B + e = \delta.$$

Both γ and ξ are closed. For any $x \in \Omega$ satisfying $\operatorname{dist}(x, \Gamma) > \delta$, define

$$H(v, s) = (1 - v)\gamma(s) + v\xi(s), \qquad v \in [0, 1].$$

This is a continuous deformation through closed curves, and

$$\|H(v, s) - x\|_2 \ge \operatorname{dist}(x, \Gamma) - \delta > 0.$$

Consequently, the winding number around x is unchanged throughout the deformation, by homotopy invariance of the degree of the normalized direction map to the circle (Hatcher, 2002, Section 2.2, p. 134). The intermediate curves need not be simple. Since γ traverses a simple closed curve once, its winding number is 1 or -1 in R and zero outside $R \cup \Gamma$. The even–odd fill of ξ therefore assigns the same label as R to x.

Finally, both the reference contour and the predicted contour lie in the stated boundary band, so the convention for labeling points on either contour cannot affect the conclusion outside that band. Every point of $R \triangle \widehat{R}$ therefore lies within distance δ of Γ. □

For masks evaluated at pixel centers, every center farther than δ from Γ receives the correct label.

For several contours, require a complete matched collection, with each pair satisfying Corollary 8 and both masks using even–odd filling. Winding parity agrees for each pair outside the union of their error bands; summing the parities modulo two confines mask disagreement to that union. This covers components and holes, but excludes omitted or unmatched contours.

The bound controls neither topology nor intersection over union: the δ-band may contain spurious components or cover a thin object.

C.4 Region containment and learned samples

For a region $R \subseteq \Omega$ containing the reference curve, the clearance used in Corollary 9 is

$$\rho = \inf_{s\in[0,L]} \operatorname{dist}(\gamma(s), \Omega \setminus R), \tag{29}$$

with $\operatorname{dist}(x, \varnothing) = +\infty$. If the Fréchet bound is strictly below ρ, there is a matching in Equation (24) whose maximum distance is below ρ. Every predicted point occurs in that matching and lies in the convex domain Ω, so a point outside R would contradict the definition of ρ.

In Corollary 10, the integer-grid samples $\bar{p}_i = BQ_B(\gamma(s_i))$ form a path satisfying Equation (2) in a mask $M \subseteq \Lambda_{B+1}$. Theorem 3 recovers this path; dividing by B gives

$$\widehat{P} = \big(Q_B(\gamma(s_0)), \ldots, Q_B(\gamma(s_J))\big), \qquad d_{\mathrm{F}}(\gamma, \Pi(\widehat{P})) \le \frac{\kappa h^2}{8} + \frac{\sqrt{2}}{2B}. \tag{30}$$

The path condition does not follow from smoothness alone: rounding may merge samples or create extra occupied neighbors.

D Visual Connectivity

D.1 Visual Encoding

Graph representation. An image consists of aligned blocks of a common constant size, with bridge marks separating crossing curve segments. Each block permutes 120 lanes indexed by the elements of S_5, the permutation group on five elements. With identity e and composition $(\sigma\tau)(k) = \sigma(\tau(k))$, the template W_σ connects incoming lane g to outgoing lane σg. The templates are distinct. An image with m blocks and a fixed endpoint frame has $N_{\mathrm{img}} = \Theta(m + 1)$ raster bits.

For $W = W_{\sigma_1} \cdots W_{\sigma_m}$, $m \geq 0$, the raster image I_W represents the undirected graph $G_W = (V_W, E_W)$ with

$$\begin{aligned} V_W &= \{(j, g) : 0 \leq j \leq m,\ g \in S_5\}, \\ E_W &= \big\{\{(j-1, g), (j, \sigma_j g)\} : 1 \leq j \leq m,\ g \in S_5\big\}. \end{aligned} \tag{31}$$

The image includes fixed markers for $s_{\mathrm{src}} = (0, e)$ and $t_{\mathrm{dst}} = (m, e)$. The language VCON_{S_5} consists of valid images for which s_{src} and t_{dst} are connected in G_W; malformed images are rejected.

Each graph edge follows its curve segment in the template; marked crossings add no edges. The path from s_{src} ends in lane $\sigma_m \cdots \sigma_1$, with the empty product equal to e. This path exists for either answer: connectivity holds exactly when its terminal is t_{dst}.

Raster encoding. The regular left action of S_5 defines a lane permutation $g \mapsto \sigma g$ for each $\sigma \in S_5$. Decompose each permutation into adjacent lane swaps, each represented by a fixed raster bridge. There are finitely many templates and 120 lanes, so their maximum height, width, and number of swaps are constants. Padding gives a common block size. Distinct curves have positive separation except at marked bridges, whose patterns specify the two separate connections. The polygonal centerlines at bridges are interpreted according to these marks.

Fix positive integers H, w, f for the raster height, block width, and total frame width, respectively. Put $L_m = f + mw$. The input is the row-major binary encoding of an $H \times L_m$ raster; hence $N_{\mathrm{img}} = HL_m$ determines m. Geometric template vertices lie in the canvas $[0, L_m] \times [0, H]$; dividing their raw horizontal and vertical coordinates by L_m and H, respectively, places them in Ω. All CoC points and boundary positions X_j, Y_g in Section 4.4 and this appendix use these normalized coordinates. Incompatible input lengths are rejected. The frame contains the endpoint markers, including coincident endpoints for $m = 0$. All dimensions and coordinates are fixed by this format. Equality tests against the 120 fixed templates recognize each block; conjunction with the frame-validity bits rejects malformed images in AC^0. The image dimensions determine the block boundaries. Each pixel of a symbol-to-block substitution depends only on its source symbol; the frame pixels are constants. The templates are distinct because different σ connect the identity lane to different outgoing lanes.

For $m \geq 1$, every vertex of G_W in an interior layer has exactly one neighbor in each adjacent layer, and each boundary vertex has degree one. Each connected component is therefore a path from layer 0 to layer m. For $m = 0$, the components are isolated vertices. Starting at lane g gives terminal lane $\sigma_m \cdots \sigma_1 g$, so in all cases

$$s_{\mathrm{src}} \longleftrightarrow t_{\mathrm{dst}} \text{ in } G_W \quad \Longleftrightarrow \quad \sigma_m \cdots \sigma_1 = e, \tag{32}$$

with the empty product equal to e. Subdividing edges into their template routes and retaining separate connections at bridges preserves this equivalence.

D.2 Inference Model and Coordinate Construction

Inference model. Circuit families and reductions are nonuniform and polynomial-size. AC^0 has constant depth with unbounded-fan-in AND/OR gates and NOT gates; TC^0 additionally permits majority gates; NC^1 has logarithmic depth and bounded-fan-in Boolean gates.

For this comparison, we require the complete next-token map, including visual preprocessing, to be a total TC^0 function with depth independent of image and prefix length. Each step receives only the N_{img}-bit image, a prefix padded to a fixed polynomial context bound, and its length; the endpoint query is fixed by the image markers. Inference is deterministic over a fixed finite vocabulary with distinct answer tokens; it starts with an empty prefix and halts at an answer token. Circuit bounds for transformers require the corresponding architecture and precision conditions (Merrill & Sabharwal, 2023a;b; Chiang, 2025).

Coordinate construction. Graph vertices (j, g) are combinatorial labels. Their rendered positions use axis-wise normalization to Ω (Appendix D.1). Let $p_j^{\partial} = (X_j, Y_{g_j}) \in \Omega$ be the point where the curve from the source meets boundary j; X_j is the normalized horizontal position and Y_g the normalized height of lane g. The CoC includes both boundary coordinates and vertices inside each

block. The boundary points obey

$$\begin{aligned} g_0 &= e, \qquad p_0^{\partial} = (X_0, Y_e), \\ g_j &= \sigma_j g_{j-1}, \qquad p_j^{\partial} = (X_j, Y_{g_j}), \quad 1 \le j \le m. \end{aligned} \tag{33}$$

The lane label g_j is recoverable from p_j^{∂}. The boundary coordinates form a subsequence of the CoC.

Pad every fixed template route to r edges by repeating its terminal vertex, for a constant $r \ge 1$. Concatenating these routes gives $p_{rj} = p_j^{\partial}$. For $0 \le t < rm$, write $t = rj + k$, where $0 \le k < r$. From the previously generated points $(p_0, \ldots, p_t)$, the decoder retrieves $p_j^{\partial} = p_{rj}$ and uses block $j + 1$ to select the next vertex p_{t+1}, applying Equation (1) at index $t + 1$. The indices j, k follow from the number of completed records. Apart from image-validity checking, each update uses one block and one earlier coordinate. The complete CoC has $K = rm + 1$ points; its final point p_m^{∂} determines the answer $\mathbf{1}[g_m = e]$. Coordinates use the fixed-width encoding of Appendix A.1.

D.3 Proof of Theorem 12

Proof. Completeness. Parallel template recognition and a balanced tree of constant-size group multiplications evaluate $\sigma_m \cdots \sigma_1$ in NC^1; testing image validity and equality to e therefore decides connectivity, including the empty instance. For hardness, use the S_5 identity-word problem, which is NC^1-hard under AC^0 reductions that output valid group words (Barrington, 1989), as in the CoT separation of Li et al. (2024, Theorem 3.5). Reversing the word is a fixed permutation of input positions and gives our multiplication convention. By Appendix D.1, this rendering is a linear-length AC^0 substitution. The path from the source ends in lane $\sigma_m \cdots \sigma_1$ by Equation (32), so the substitution preserves the answer and establishes hardness on valid images.

Logarithmic-token limitation. Suppose a solver uses at most $T_{\mathrm{tok}}(N_{\mathrm{img}}) = O(\log N_{\mathrm{img}})$ intermediate tokens. A fixed vocabulary gives only polynomially many candidate transcripts ending in acceptance, with at most $T_{\mathrm{tok}}(N_{\mathrm{img}})$ preceding tokens and no earlier answer. Following the transcript-enumeration argument of Li et al. (2024, Section 3.3), substitute each candidate prefix into the next-token circuit and check all predictions in parallel. Conjoining the checks for each candidate and disjoining over candidates preserves TC^0 depth and polynomial size; determinism makes this circuit accept exactly when the solver accepts. Composing it with the valid-image reduction above gives $\mathsf{NC}^1 \subseteq \mathsf{TC}^0$, contradicting the assumption.

Traversal and decision. The image-validity test rejects malformed inputs in AC^0. On a valid image, initialize $p_0 = p_0^{\partial} = (X_0, Y_e)$. At boundary $j < m$, the boundary record p_j^{∂} identifies j and g_j; recognition of block $j + 1$ gives σ_{j+1}. Induction under Equation (33) yields $g_j = \sigma_j \cdots \sigma_1$. By Equation (32), the decision $\mathbf{1}[g_m = e]$ is correct. If $g_m \ne e$, the entire source component ends at $(m, g_m) \ne t_{\mathrm{dst}}$, so the same traversal establishes non-connectivity.

CoC construction and encoding. Use the r-edge padded template routes fixed in Appendix D.2. At boundary j, the table indexed by (σ_{j+1}, g_j) supplies the next r vertices. Translation into block $j+1$ in raster coordinates, normalization as in Appendix D.1, and concatenation yield exactly the curve traversed from the source. The resulting CoC $P = (p_0, \ldots, p_{rm})$ satisfies $p_{rj} = p_j^{\partial}$: the last vertex of each block route is $p_{r(j+1)} = p_{j+1}^{\partial}$. This preserves the boundary recurrence and final decision. For $m = 0$, the CoC contains only p_0 and the answer is positive.

Let D be a common integer denominator for the fixed template vertices, and use the raster width $L_m = f + mw = \Theta(m + 1)$ and constant height H from Appendix D.1. After normalization, $B_m = DHL_m = O(N_{\mathrm{img}})$ represents every translated vertex exactly on $\mathcal{G}_{B_m}$. By Appendix A.1, each coordinate record has $2\ell_{B_m,b} + O(1) = O(\log N_{\mathrm{img}})$ tokens. The $rm + 1 = O(N_{\mathrm{img}})$ records and final decision token require $O(N_{\mathrm{img}} \log N_{\mathrm{img}})$ tokens and a polynomial context.

Per-token computation. Fix the raster bit length $N_{\mathrm{img}} = \Theta(m + 1)$. Lengths incompatible with the raster format are rejected. Otherwise, N_{img} determines m, the grid denominator, and the padded coordinate-field lengths; the format requires no instance-dependent header. Prefix length determines the current record and radix-symbol phase, including the initial coordinate and final answer. Because every block contributes r records, prefix length also determines the position of the preceding boundary record. Before generating p_{t+1}, for $0 \le t < rm$, this record has index

$$\iota(t) = r\lfloor t/r \rfloor \le t, \qquad p^{\partial}_{\lfloor t/r \rfloor} = p_{\iota(t)} \in \{p_0, \ldots, p_t\}. \tag{34}$$

Writing $t = rj + k$ with $0 \le k < r$, the next point is vertex $k + 1$ of the translated route for (σ_{j+1}, g_j). Positional tables select $\iota(t)$, $k + 1$, and the radix-symbol phase. Every radix symbol uses that completed boundary record and the current block, even when p_t lies inside the block. This applies Equation (1) at index $t+1$, using the image and previously generated points without separate lane memory. The final boundary record determines the answer.

There are polynomially many phase, boundary-coordinate, and template combinations. Phase equality tests use $O(\log N_{\mathrm{img}})$ bit literals, and coordinate equality tests use $O(\log N_{\mathrm{img}})$ constant-size radix-symbol equality tests. Template equality tests involve only a constant number of pixels. Each output-code bit is an OR over the combinations that prescribe that bit, using phase-selected prefix positions. The fixed table includes translated coordinate radix symbols, delimiters, the initial record, and the final lane-equality decision. Consequently the complete next-token map is a constant-depth polynomial-size Boolean circuit, using generated tokens as state (Merrill & Sabharwal, 2024; Li et al., 2024). Invalid images, out-of-range prefix lengths, and malformed selected records receive a fixed rejecting answer, making the map total on all padded prefixes. No test of the prefix's historical correctness is required; induction gives the prescribed outputs on generated prefixes.

Lemma A.1 realizes these circuits and accesses the addressed boundary record with constant depth per token and polynomial resources. The construction therefore meets the complete per-step bound of Appendix D.2. □

D.4 Proof of Proposition 13

Proof. The raster length fixes m, the coordinate precision, and all record positions. Check image validity, the complete format, and $p_0 = (X_0, Y_e)$. For every $0 \le t < rm$ in parallel, write $t = rj+k$ with $0 \le k < r$. Check that $p_{rj} = (X_j, Y_g)$ for some lane g, recognize template σ_{j+1}, and compare p_{t+1} with vertex $k + 1$ of its fixed translated route from lane g. Each coordinate equality is a conjunction of $O(\log N_{\mathrm{img}})$ constant-size radix-symbol equality tests. Each table lookup is an OR of polynomially many conjunctions. AND all checks and require $a = \mathbf{1}[p_{rm} = (X_m, Y_e)]$. The depth is constant and size polynomial. Induction from the checked start shows that any accepted CoC follows the prescribed vertices of the curve from the source, proving soundness; the canonical CoC passes every check, proving completeness. For $m = 0$, only the start, format, and positive answer are required. □

E TRAINING HYPERPARAMETERS

Table 3: Training hyperparameters.

Setting	Value
Supervised fine-tuning	
LoRA rank / scaling	$r = 32$, $\alpha = 32$
Precision	bf16
Optimizer	8-bit AdamW
Learning rate	10^{-4}
Schedule	Linear decay; 3% warmup
Weight decay	0.001
Epochs	2
Effective batch size	8
Maximum sequence length	4,096 tokens
Random seed	3407
Reinforcement learning	
Objective	GRPO; clip 0.2; KL weight $\beta = 0$
Sampling	$G = 8$; temperature 1.0; top-$p = 1.0$
Optimizer / schedule	8-bit AdamW
Learning rate	5×10^{-6}
Per-device batch $\times$ accumulation	4×4
Reward weights $(\lambda_{\mathrm{F}}, \lambda_{\mathrm{fmt}}, \lambda_{\mathrm{ans}})$	$(0.5, 0.2, 1.0)$; $\lambda_{\mathrm{ans}} = 0$ for drawing/writing
Completion budget (tokens)	3,072

F COC PROMPTING

CoC prompting retains the original task question and adds explicit curve tracing as an intermediate reasoning step. The added instruction takes the following form:

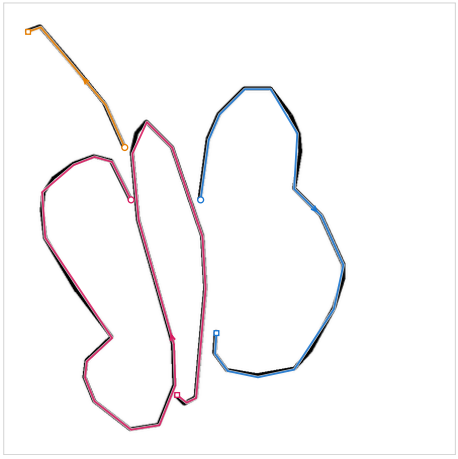
> First reason by tracing the relevant structure in the image as a sequence of (x,y) coordinates: output `<CoC>(x1,y1),(x2,y2),...</CoC>` using integer coordinates on a 0-1000 grid, where x is measured from the left edge and y from the top edge of the image, following the structure point by point in order. Then, based on that trace, give your final answer as `<answer>...</answer>`.

G Qualitative Showcase

G.1 Quick, Draw!: sketch reconstruction

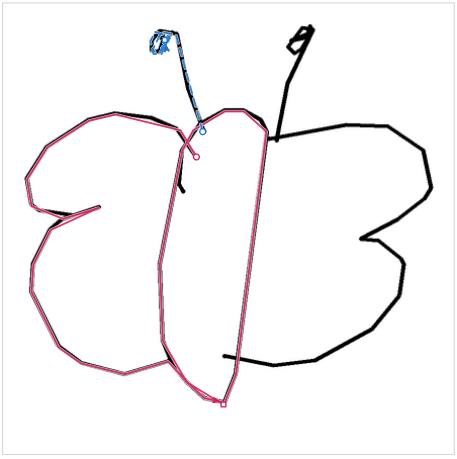
Qwen3.5 2B + CoC SFT + RL

Best Q109



Query: The image shows a hand-drawn sketch. Draw it: give the pen trajectory that draws this sketch, stroke by stroke, in drawing order.
Fréchet distance: 0.0054
Response: (282,436),(240,352). . .
CoC: 3 sequences, 42 points.

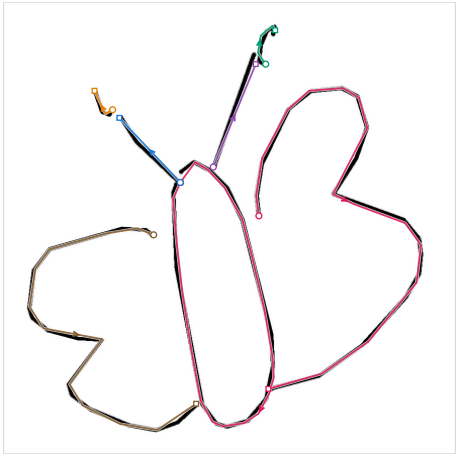
Worst Q110



Query: The image shows a hand-drawn sketch. Draw it: give the pen trajectory that draws this sketch, stroke by stroke, in drawing order.
Fréchet distance: 0.8956
Response: (429,341),(394,282). . .
CoC: 2 sequences, 130 points. 1 unterminated; available prefixes dashed.

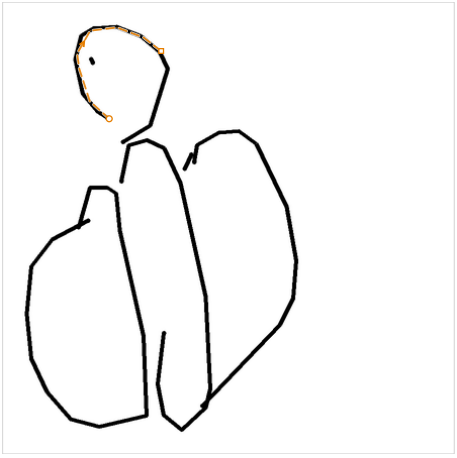
Gemini 3.8 Flash with CoC prompting

Best Q117



Query: The image shows a hand-drawn sketch. Draw it: give the pen trajectory that draws this sketch, stroke by stroke, in drawing order.
Fréchet distance: 0.0118
Response: (391,399),(424,354). . .
CoC: 7 sequences, 61 points.

Worst Q118

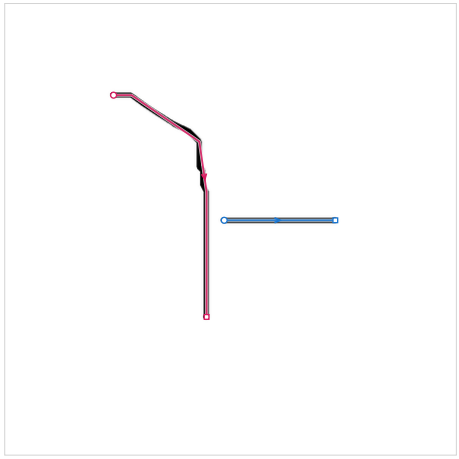


Query: The image shows a hand-drawn sketch. Draw it: give the pen trajectory that draws this sketch, stroke by stroke, in drawing order.
Fréchet distance: 1.0000
Response: “‘json [{"point": [236, 258]}, {"point": [193, 219]}, {"point":. . .
CoC: 3 sequences, 8 points. 1 unterminated; available prefixes dashed.

G.2 OMNIGLOT: CHARACTER RECONSTRUCTION

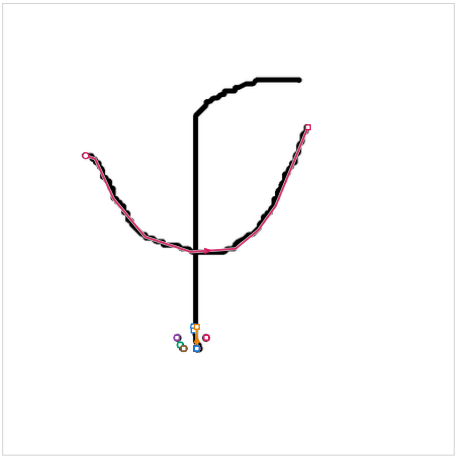
Qwen3.5 2B + CoC SFT + RL

Best Q089



Query: The image shows a handwritten character. Write it: give the pen trajectory that draws this character, stroke by stroke, in writing order.
Fréchet distance: 0.0000
Response: (241,203),(281,203). . .
CoC: 2 sequences, 7 points.

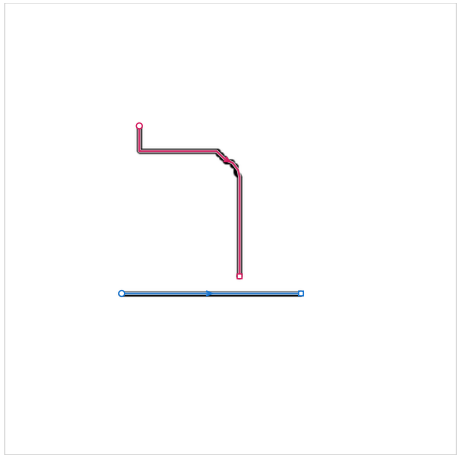
Worst Q090



Query: The image shows a handwritten character. Write it: give the pen trajectory that draws this character, stroke by stroke, in writing order.
Fréchet distance: 0.8117
Response: (184,337),(205,344). . .
CoC: 44 sequences, 55 points.

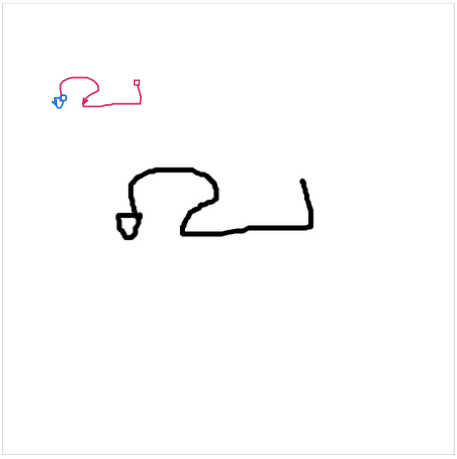
GPT-5.6 Sol with CoC prompting

Best Q103



Query: The image shows a handwritten character. Write it: give the pen trajectory that draws this character, stroke by stroke, in writing order.
Fréchet distance: 0.0042
Response: (298,272),(298,328). . .
CoC: 2 sequences, 10 points.

Worst Q104

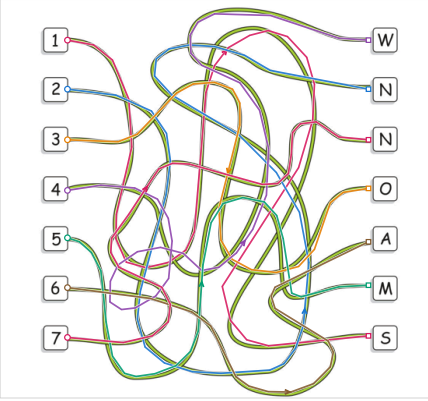


Query: The image shows a handwritten character. Write it: give the pen trajectory that draws this character, stroke by stroke, in writing order.
Fréchet distance: 0.7293
Response: (132,211),(130,202). . .
CoC: 2 sequences, 34 points.

G.3 BABYVISION: CONNECT THE LINES

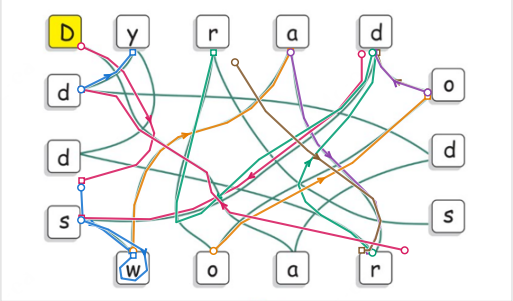
Qwen3.5 2B + CoC SFT + RL

Correct Q147



Query: Which letter on the right is connected to which number on the left by a green line? (The answer format should be similar to 1-W,2-N,3-N,4-O,5-A,6-M,7-S)
Response: 1-S,2-N,3-O,4-W,5-M,6-A,7-N
Ground Truth: 1-S,2-N,3-O,4-W,5-M,6-A,7-N
CoC: 7 sequences, 168 points.

Incorrect Q148



Query: Starting from the letters in the yellow area, follow the lines, and combine the connected letters in sequence to form 3 complete English words.
Response: dd,adw,osrdodaa
Ground Truth: Daddy,draws,doors
CoC: 14 sequences, 92 points.

GPT-5.6 Luna with CoC prompting

Correct Q014



Query: Into which color bin would "plastic cup," " newspaper," and "apple core" go respectively in the picture? ("plastic cup," "newspaper," and "apple core" are A, B, C respectively; answer format is like: A-yellow, B-blue)
Response: A-blue, B-yellow, C-green
Ground Truth: A - Blue, B - Yellow, C - Green
CoC: 3 sequences, 82 points.

Incorrect Q015

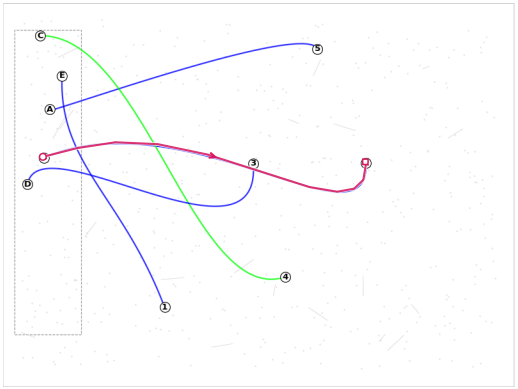


Query: There are three animal patterns at the top of the picture and three environment patterns below, which are connected by lines in order. What is the connection method? (Assume the animal patterns above are A, B, C, and the environment patterns below are 1, 2, 3. The answer format is like: A-1, B-2, C-3)
Response: A-1, B-2, C-3
Ground Truth: A-2,B-3,C-1
CoC: 3 sequences, 63 points.

G.4 TRACER-BENCH: NONSEMANTIC TRACING

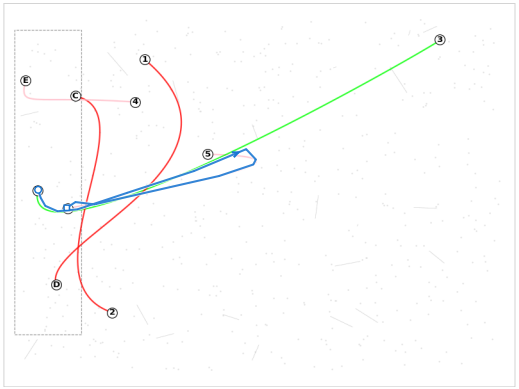
Qwen3.5 2B + CoC SFT + RL

Correct Q125



Query: Starting from the letter B, which number does the line end at?
Response: {2}
Ground Truth: 2
CoC: 1 sequence, 11 points.

Incorrect Q126

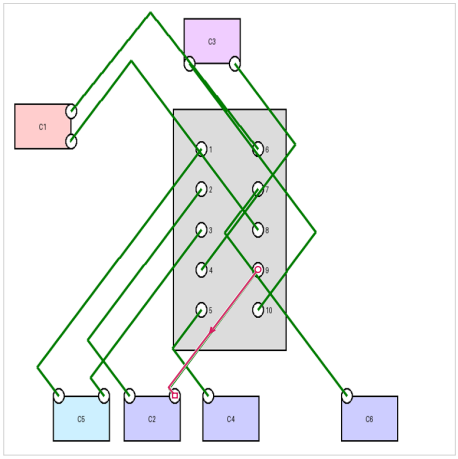


Query: Starting from the letter B, which number does the line end at?
Response: {1}
Ground Truth: 3
CoC: 1 sequence, 14 points.

G.5 TUNNEL VISION: SINGLE COLOR

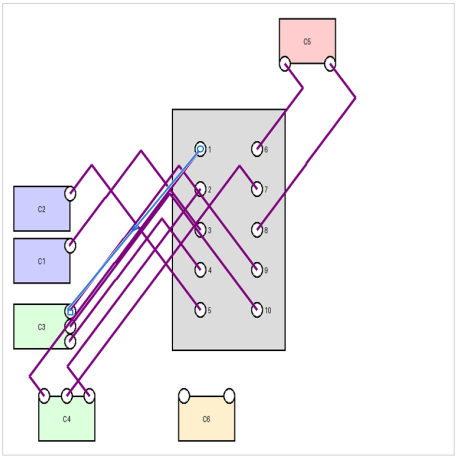
Qwen3.5 2B + CoC SFT

Correct Q024



Query: Which component does the wire from port 9 on the breadboard, which is the gray rectangle with numbered ports, connect to?
Response: C2
Ground Truth: C2
CoC: 1 sequence, 4 points.

Incorrect Q025

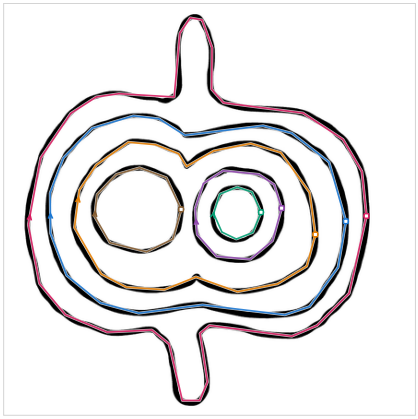


Query: Which component does the wire from port 1 on the breadboard, which is the gray rectangle with numbered ports, connect to?
Response: C3
Ground Truth: C4
CoC: 1 sequence, 4 points.

G.6 CurveBench: containment trees

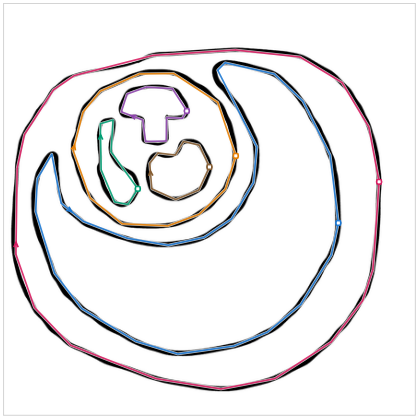
Qwen3.5 2B + CoC SFT + RL

Correct Q056



Query: Determine the containment tree as an edge list [[parent,child],...] where each edge means the child region is directly inside the parent region. Region 0 is the unbounded outside region.
Response: [[0,1],[1,2],[2,3],[3,4],[3,6],[4,5]]
Ground Truth: [[0,1],[1,2],[2,3],[3,4],[3,6],[4,5]]
CoC: 6 sequences, 99 points.

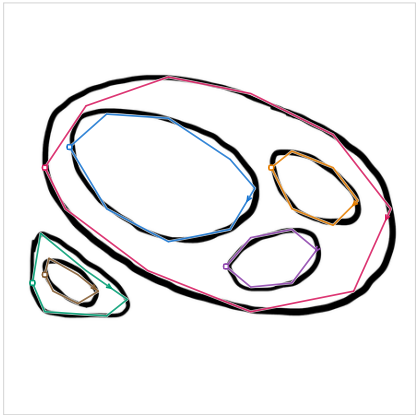
Incorrect Q057



Query: Determine the containment tree as an edge list [[parent,child],...] where each edge means the child region is directly inside the parent region. Region 0 is the unbounded outside region.
Response: [[0,1],[1,2],[2,3],[3,4],[3,5],[3,6]]
Ground Truth: [[0,1],[1,2],[1,6],[2,3],[2,4],[2,5]]
CoC: 6 sequences, 88 points.

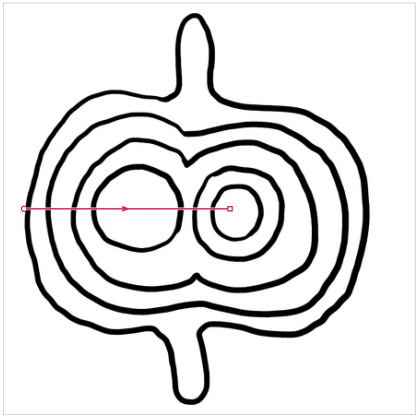
Gemini 3.1 Pro with CoC prompting

Correct Q068



Query: Determine the containment tree as an edge list [[parent,child],...] where each edge means the child region is directly inside the parent region. Region 0 is the unbounded outside region.
Response: [[0, 1], [1, 2], [1, 3], [1, 4], [0, 5], [5, 6]]
Ground Truth: [[0,1],[0,3],[1,2],[3,4],[3,5],[3,6]]
CoC: 6 sequences, 48 points.

Incorrect Q069



Query: Determine the containment tree as an edge list [[parent,child],...] where each edge means the child region is directly inside the parent region. Region 0 is the unbounded outside region.
Response: [[0, 1], [1, 2], [2, 3], [3, 4], [3, 5]]
Ground Truth: [[0,1],[1,2],[2,3],[3,4],[3,6],[4,5]]
CoC: 1 sequence, 6 points.