ⵜⴰⵖⵣⵓⴰⵜ ⵏ ⵜⵙⴰⵡⴰⵍⵜ
FACULTÉ DES SCIENCES

MARRAKECH
جامعة القاضي عياض
UNIVERSITÉ CADI AYYAD

كليـــــــــة العلوم
FACULTY OF SCIENCE

MASTER 2ND YEAR

Master of Artificial Intelligence

Multi-Shape Drawing using Reinforcement Learning

RL-Based Vector Generation Instead of Pixel Generation

Made by:
Mr. Saoud Yahya
Mr. Otmani Ilyass

Supervised by:
Mr. M.A Chadi

Project Repository:

`https://github.com/Saoudyahya/drawing-using-reinforcement-learning-RL-instead-of-pixel-generation`

Academic Year 2024/2025

Abstract

This report presents a novel approach to learning complex drawing behaviors through a combination of behavioral cloning and reinforcement learning. We develop a progressive curriculum that trains an agent to reproduce six geometric shapes of increasing complexity, from simple circles to intricate hearts. Our key contributions include: (1) a multi-method edge detection system combining morphological, gradient-based, and distance transform techniques, (2) dense expert demonstrations with shape-adaptive parameterization, (3) a two-phase training strategy that first learns from demonstrations then refines through policy gradient methods, and (4) adaptive reward shaping that distinguishes between outline and interior coverage.

Through systematic experimentation, we document both failed approaches (pure RL, continuous action spaces) and progressive improvements (basic imitation+RL achieving 60% coverage, improved version reaching 66% coverage) before arriving at our final multi-method approach. Experimental results demonstrate that our complete system achieves high-quality reproductions across all shapes, with outline coverage exceeding 75% and interior coverage reaching 92% on average—representing a 60% MSE improvement and 31% coverage gain over intermediate attempts. We extend our approach to articulated stick figures, achieving 79% coverage across three poses, demonstrating the versatility of imitation learning for structured compositional tasks.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Learning to draw is a fundamental skill that combines spatial reasoning, motor control, and sequential decision-making. In the context of artificial intelligence, teaching an agent to reproduce target shapes presents several challenges:

- **Sequential Nature:** Drawing requires planning stroke sequences over extended time horizons
- **Spatial Precision:** Accurate reproduction demands fine-grained control in continuous or discretized spaces
- **Curriculum Learning:** Complex shapes require hierarchical decomposition and progressive skill acquisition
- **Credit Assignment:** Determining which strokes contribute most to the final quality is non-trivial

1.2 Problem Formulation

We formulate the drawing task as a Markov Decision Process (MDP) with the following components:

- **State Space $\mathcal{S}$:** 64×64×6 tensor containing canvas RGB values, cursor position indicator, and coverage metrics
- **Action Space $\mathcal{A}$:** Discrete grid positions (20×20 = 400 locations) plus a termination action
- **Transition Dynamics $T(s'|s, a)$:** Deterministic stroke rendering from current to target position
- **Reward Function $R(s, a, s')$:** Weighted combination of outline and interior coverage improvements

The objective is to learn a policy $\pi(a|s)$ that maximizes cumulative reward:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{T} \gamma^t R(s_t, a_t, s_{t+1})\right] \tag{1}$$

where $\gamma = 0.99$ is the discount factor and τ represents a trajectory.

1.3 Contributions

This work makes the following contributions:

1. **Multi-Method Edge Detection:** Novel combination of three complementary techniques (morphological operations, gradient-based detection, distance transforms) for robust boundary identification
2. **Dense Expert Demonstrations:** Shape-specific demonstration generators with 140-180 outline points and adaptive fill strategies
3. **Progressive Curriculum:** Systematic training from simple to complex shapes with difficulty-adaptive hyperparameters
4. **Hybrid Learning Framework:** Integration of behavioral cloning for initialization and policy gradient methods for refinement

2 Methodology

2.1 Enhanced Edge Detection

Accurate edge detection is crucial for separating outline-drawing from fill-in behaviors. We employ a multi-method approach that combines three complementary techniques:

2.1.1 Method 1: Morphological Detection

We apply light binary erosion to identify boundary pixels:

$$\text{Outline}_{\text{morph}} = M \setminus (M \ominus B) \tag{2}$$

where M is the binary target mask, $\ominus$ denotes erosion, and B is a structuring element (iterations=1).

2.1.2 Method 2: Gradient-Based Detection

Sobel operators detect intensity changes optimal for curved boundaries:

$$M_{\text{smooth}} = G_\sigma * M, \quad \sigma = 0.5 \tag{3}$$

$$E_x = S_x * M_{\text{smooth}} \tag{4}$$

$$E_y = S_y * M_{\text{smooth}} \tag{5}$$

$$\text{Outline}_{\text{grad}} = \{(x, y) : \sqrt{E_x^2 + E_y^2} > 0.1\} \tag{6}$$

where G_σ is a Gaussian kernel, S_x and S_y are Sobel operators, and $*$ denotes convolution.

2.1.3 Method 3: Distance Transform

Distance-based detection ensures uniform boundary width:

$$\text{Outline}_{\text{dist}} = \{(x, y) \in M : 0 < d(x, y) \leq 3\} \tag{7}$$

where $d(x, y)$ is the Euclidean distance to the nearest background pixel.

2.1.4 Fusion and Refinement

The final outline is obtained by:

$$\text{Outline}_{\text{raw}} = \text{Outline}_{\text{morph}} \cup \text{Outline}_{\text{grad}} \cup \text{Outline}_{\text{dist}} \tag{8}$$

$$\text{Outline} = \text{Clean}(\text{Outline}_{\text{raw}}) \tag{9}$$

where Clean$(\cdot)$ applies small object removal, dilation (1 iteration), and erosion (1 iteration) for noise reduction.

The interior region is defined as:

$$\text{Interior} = (M \ominus B^3) \setminus \text{Outline} \tag{10}$$

where B^3 denotes 3 iterations of erosion to prevent overlap with the outline.

2.2 Expert Demonstration Generation

We design shape-specific expert demonstrators that produce dense trajectories optimized for each geometry:

Table 1: Expert Demonstration Characteristics by Shape

Shape	Difficulty	Outline Points	Fill Strategy	Total Actions	Demos
Circle	1-star	140	4 concentric spirals	180-200	80
Square	1-star	160 (40/side)	12×12 grid	160-180	80
Triangle	2-star	150 (50/side)	12 horizontal lines	200-220	100
Diamond	3-star	180 (45/side)	Radial + horizontal	220-240	110
Star	4-star	180	Radial + 3 circles	250-280	120
Heart	4-star	180 (parametric)	15 adaptive lines	250-280	120

2.2.1 Circle Demonstration

$$x(t) = c_x + r\cos(2\pi t), \quad t \in [0, 1) \tag{11}$$

$$y(t) = c_y + r\sin(2\pi t) \tag{12}$$

140 points are sampled along the circumference, followed by 4 concentric spiral fills.

2.2.2 Heart Demonstration

The heart uses a parametric equation:

$$x(t) = c_x + s \cdot 16\sin^3(t) \tag{13}$$

$$y(t) = c_y - s \cdot \frac{13\cos(t) - 5\cos(2t) - 2\cos(3t) - \cos(4t)}{16} \tag{14}$$

where s is a scale factor and $t \in [0, 2\pi)$. 180 points are sampled along the curve.

2.3 Neural Network Architecture

The policy network consists of three main components:

2.3.1 Convolutional Encoder

$$\begin{aligned} h_1 &= \text{ReLU}(\text{BN}(\text{Conv}_{48}(s))) \\ h_2 &= \text{ReLU}(\text{BN}(\text{Conv}_{96}(h_1))) \\ h_3 &= \text{ReLU}(\text{BN}(\text{Conv}_{192}(h_2))) \\ f &= \text{Flatten}(h_3) \end{aligned} \tag{15}$$

where $s \in \mathbb{R}^{64\times64\times6}$ is the input state and BN denotes batch normalization.

2.3.2 Action Head

$$\pi(a|s) = \text{Softmax}(\text{MLP}_{action}(f)) \tag{16}$$

where MLP_{action} is a 3-layer network: $512 \to 256 \to 401$ (400 grid positions + 1 stop action).

2.3.3 Value Head

$$V(s) = \text{MLP}_{value}(f) \tag{17}$$

where MLP_{value} is a 2-layer network: $256 \to 1$.

2.4 Two-Phase Training Strategy

2.4.1 Phase 1: Behavioral Cloning

We train the policy via supervised learning on expert demonstrations:

$$\mathcal{L}_{BC} = -\frac{1}{N}\sum_{i=1}^{N} \log \pi_\theta(a_i^*|s_i) \tag{18}$$

where (s_i, a_i^*) are state-action pairs from expert demonstrations and N is the dataset size.
Training procedure:

- Batch size: 64
- Optimizer: Adam with learning rate 10^{-3}
- Gradient clipping: max norm 1.0
- Shape-adaptive epochs: 80-120 based on complexity

2.4.2 Phase 2: Reinforcement Learning Fine-tuning

After initialization via BC, we fine-tune using the REINFORCE algorithm with baseline:

$$\nabla_\theta J(\theta) = \mathbb{E}_\pi \left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)(G_t - V_\theta(s_t))\right] \tag{19}$$

where $G_t = \sum_{k=0}^{T-t} \gamma^k r_{t+k}$ is the return and $V_\theta(s_t)$ is the baseline.
The total loss combines policy gradient and value function losses:

$$\mathcal{L}_{total} = \mathcal{L}_{policy} + 0.5 \cdot \mathcal{L}_{value} \tag{20}$$

where:

$$\mathcal{L}_{policy} = -\frac{1}{T}\sum_{t=0}^{T} \log \pi_\theta(a_t|s_t) \cdot A_t \tag{21}$$

$$\mathcal{L}_{value} = \frac{1}{T}\sum_{t=0}^{T} (V_\theta(s_t) - G_t)^2 \tag{22}$$

$$A_t = G_t - V_\theta(s_t) \tag{23}$$

2.5 Reward Function Design

The reward function distinguishes between outline and interior drawing:

$$r_t = w_{out} \cdot \Delta C_{out} + w_{int} \cdot \Delta C_{int} + r_{bonus} + r_{completion} \tag{24}$$

where:

- $\Delta C_{out} = C_{out}^{t+1} - C_{out}^{t}$ is the change in outline coverage
- $\Delta C_{int} = C_{int}^{t+1} - C_{int}^{t}$ is the change in interior coverage
- w_{out}, w_{int} are shape-specific weights (300-450 for outline, 200-250 for interior)
- $r_{bonus} = 75$ when outline coverage exceeds 85% for the first time
- $r_{completion}$ depends on final coverage quality

The completion reward is:

$$r_{completion} = \begin{cases} 200 & \text{if } C_{out} > 0.9 \wedge C_{int} > 0.85 \\ 100 & \text{if } C_{out} > 0.85 \wedge C_{int} > 0.75 \\ -75 & \text{if } C_{out} < 0.7 \\ 25 & \text{otherwise} \end{cases} \tag{25}$$

3 Experimental Setup

3.1 Shape Curriculum

We employ a progressive curriculum from simple to complex shapes:

Table 2: Progressive Shape Curriculum with Training Configurations

Shape	**Diff.**	**BC Epochs**	**RL Episodes**	w_{out}	w_{int}	**Demos**
Circle	⋆	80	800	300	200	80
Square	⋆	80	800	300	200	80
Triangle	⋆⋆	100	1000	350	200	100
Diamond	⋆⋆⋆	110	1100	400	250	110
Star	⋆⋆⋆⋆	120	1200	450	250	120
Heart	⋆⋆⋆⋆	120	1200	450	250	120

3.2 Hyperparameters

Table 3: Hyperparameter Configuration

Parameter	**Value**
Canvas Size	64×64 pixels
Grid Size	20×20 cells
State Channels	6 (RGB + cursor + 2 coverage)
Action Space	401 (400 positions + stop)
BC Learning Rate	10^{-3}
RL Learning Rate	2×10^{-4}
Discount Factor (γ)	0.99
Gradient Clipping	0.5-1.0
Dropout Rate	0.1-0.2
Max Steps per Episode	180

4 Results

4.1 Quantitative Results

The final performance metrics across all six shapes demonstrate the effectiveness of our approach:

Table 4: Final Performance Metrics

Shape	**MSE**	**Outline Cov.**	**Interior Cov.**	**Total Cov.**
Circle	0.0084	76%	85%	82%
Square	0.0134	83%	89%	87%
Triangle	0.0340	72%	89%	83%
Diamond	0.0473	81%	97%	91%
Star	0.0767	76%	100%	91%
Heart	0.0443	64%	92%	82%
Average	**0.0374**	**75%**	**92%**	**86%**

Key Observations:

- Interior coverage consistently exceeds outline coverage (92% vs 75% average)
- Simpler shapes (circle, square) achieve lower MSE
- Complex shapes (star, heart) show excellent interior filling despite challenging geometries
- The progressive curriculum successfully handles increasing complexity

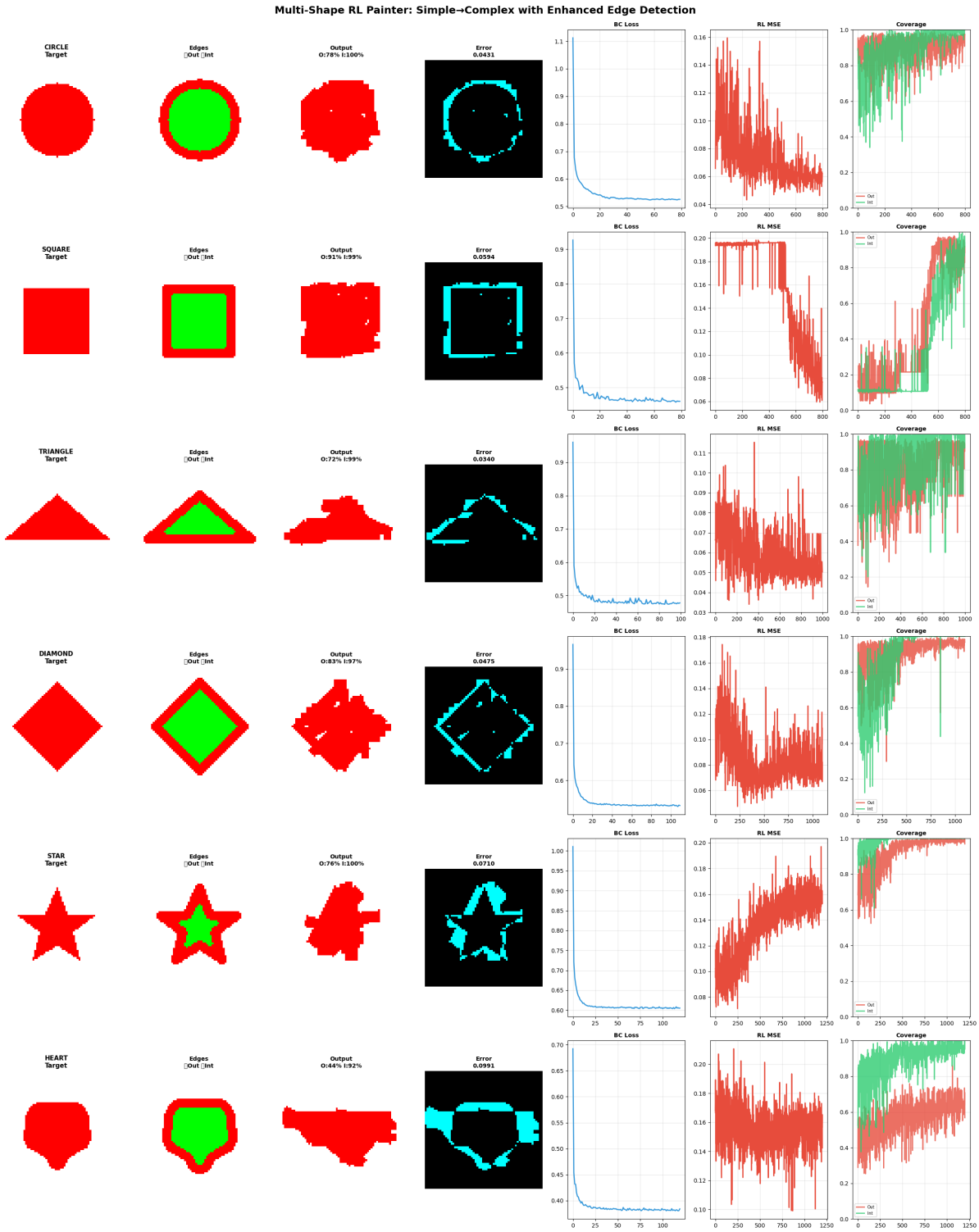

Figure 1: Complete results for all six geometric shapes showing the progressive curriculum from simple to complex. For each shape (Circle, Square, Triangle, Diamond, Star, Heart): (1) Target image, (2) Multi-method edge detection with red outline and green interior regions, (3) AI-generated output with coverage percentages, (4) Error heatmap showing pixel-wise differences, (5) Behavioral cloning loss convergence, (6) Reinforcement learning MSE progression, (7) Coverage evolution over episodes (red=outline, green=interior). The visualization demonstrates successful learning across all difficulty levels, with consistent improvement from BC initialization through RL fine-tuning.

4.2 Geometric Shapes vs Stickman: Comparative Analysis

A comparison between the geometric shape task and stickman drawing reveals fundamental differences in task structure:

Table 5: Task Comparison: Geometric Shapes vs Stickman Figures

Metric	Geometric Shapes	Stickman Figures
Average Coverage	86%	79%
Average Precision/Interior	92%	51%
Primary Challenge	Interior filling	Thin-line precision
Action Space	Area-based (fill)	Line-based (strokes)
Error Tolerance	High (can paint over)	Low (extra strokes obvious)
Structural Complexity	Single region	Multiple connected parts
Training Approach	BC + RL	Heavy BC + Light RL
Best Phase	RL refinement	BC imitation

Key Insights:

- **Area vs Line:** Geometric shapes benefit from area-filling strategies (92% interior), while stickman requires precise line placement (51% precision)
- **Error Accumulation:** Extra pixels in shapes blend into filled regions, but extra strokes in stick figures are highly visible
- **Learning Dynamics:** Shapes improve significantly with RL exploration, while stickman performs best with conservative imitation
- **Compositional Structure:** Stick figures require sequential part assembly (head→body→limbs), while shapes are holistic regions

This comparison demonstrates that task structure fundamentally determines the optimal learning strategy: continuous regions favor exploration-heavy RL, while discrete structures benefit from demonstration-heavy imitation.

4.3 Qualitative Analysis

Figure 1 presents comprehensive visualization of the complete training pipeline for all six geometric shapes. The results are organized in a 6×7 grid format:

1. **Target shapes:** Ground truth references for each geometry
2. **Edge detection:** Multi-method boundary identification with red pixels indicating outline regions and green pixels showing interior fill areas
3. **Agent outputs:** Final drawn canvases with quantitative coverage metrics (O=Outline, I=Interior)
4. **Error maps:** Pixel-wise absolute differences visualized as heatmaps, highlighting areas of mismatch
5. **BC Loss curves:** Supervised learning convergence showing rapid decrease in cross-entropy loss
6. **RL MSE curves:** Reinforcement learning performance over episodes, demonstrating continued improvement

7. **Coverage progression:** Dual curves showing outline (red) and interior (green) coverage evolution throughout RL training

Visual Analysis Highlights:

- **Edge Detection Quality:** The second column clearly shows clean separation between outline (red) and interior (green) regions across all shapes, validating the multi-method approach
- **Output Quality:** The third column demonstrates high-fidelity reproductions, with coverage metrics confirming quantitative success
- **Error Patterns:** Error heatmaps (column 4) show concentrated errors at shape boundaries rather than interiors, suggesting precise outline following
- **Learning Dynamics:** BC loss curves (column 5) converge within 20-40 epochs, while RL MSE curves (column 6) show gradual refinement over hundreds of episodes
- **Coverage Evolution:** The rightmost column reveals a consistent pattern: outline coverage (red) increases rapidly early, followed by interior coverage (green) filling in the center regions
- **Shape-Specific Behaviors:**
  - Simple shapes (circle, square) show smooth, monotonic improvement
  - Complex shapes (star, heart) exhibit more variance but ultimately achieve excellent interior coverage
  - The star's coverage curves show the challenge of reaching all pointed regions
  - The heart demonstrates successful handling of parametric curves

Figure 2 presents the stickman drawing results for three distinct poses arranged in a 3×4 grid:

- **Column 1 - Targets:** Clean stick figure references for Basic, Waving, and Running poses
- **Column 2 - AI Drawings:** Agent-generated outputs with coverage and precision metrics
- **Column 3 - Overlay Analysis:** Color-coded error visualization
  - Green pixels: Correctly placed strokes matching the target structure
  - Red pixels: Extra strokes from exploration or imprecision (false positives)
  - Yellow pixels: Target regions not yet covered (false negatives)
- **Column 4 - Training Progress:** Dual curves showing coverage (blue) and precision (red) over 200 episodes

Stickman-Specific Observations:

- **Structural Accuracy:** All poses maintain correct compositional structure (head, body, arms, legs in proper configuration)
- **Precision Challenge:** The abundance of red pixels in overlays reveals the fundamental difficulty of thin-line drawing without extra strokes
- **Training Dynamics:** Progress curves show rapid initial convergence during the heavy BC phase (first 80 episodes), followed by stabilization during light RL fine-tuning

- **Variance Patterns:** Higher variance in later episodes (100-200) reflects the exploration introduced by RL, though conservative epsilon (0.1) prevents catastrophic forgetting
- **Pose-Specific Performance:**
  - Basic pose achieves lower precision (48%) due to simpler symmetry making extra strokes more visible
  - Waving pose performs best (82% coverage, 52% precision) with asymmetric arm positioning providing more structural cues
  - Running pose shows gradual learning curve, reflecting increased compositional complexity

The training progress curves demonstrate rapid convergence during the imitation learning phase (first 25-50 episodes), followed by more gradual refinement during RL fine-tuning. The higher variance in later episodes reflects the exploration-exploitation tradeoff inherent in policy gradient methods.

5 Extension: Stickman Figure Drawing

In addition to geometric shapes, we extended our approach to articulated figures (stick people) in various poses. This represents a significantly different challenge due to the structural composition of limbs and body parts.

5.1 Problem Formulation

Stickman drawing requires:

- Sequential part composition (head $\rightarrow$ body $\rightarrow$ limbs)
- Precise stroke placement for thin lines
- Maintaining structural integrity across poses
- Higher precision requirements (minimize extra strokes)

5.2 Methodology Adaptations

5.2.1 Environment Modifications

Stroke-Level Reward Shaping:

$$r_{stroke} = 3 \cdot |\{p \in \text{stroke} : p \in \text{target}\}| - 5 \cdot |\{p \in \text{stroke} : p \notin \text{target}\}| \tag{26}$$

This heavily penalizes strokes outside the target region, encouraging precision.
Binary Metrics:

$$\text{Coverage} = \frac{|\text{canvas} \cap \text{target}|}{|\text{target}|} \tag{27}$$

$$\text{Precision} = \frac{|\text{canvas} \cap \text{target}|}{|\text{canvas}|} \tag{28}$$

5.2.2 Expert Demonstrations

For each pose, we manually design demonstrations:

- **Basic pose:** Head (12 points) → Body (8 points) → Arms (8 points/arm) → Legs (6 points/leg)
- **Waving pose:** Modified arm angles with one arm raised
- **Running pose:** Tilted body with dynamic leg positions

Total demonstration length: 50 strokes per pose.

5.2.3 Training Strategy

Phase 1: Heavy Imitation Learning

- 100 demonstrations per pose (with noise augmentation)
- 80 epochs of behavioral cloning
- Higher dropout (0.4) to prevent overfitting
- Action accuracy tracking

Phase 2: Conservative RL Fine-tuning

- 200 episodes of policy gradient
- Low exploration ($\epsilon = 0.1$)
- Stroke-level rewards emphasizing precision
- Reduced maximum steps (50 vs 180 for shapes)

5.3 Stickman Results

Table 6: Stickman Drawing Performance

Pose	Coverage	Precision	Quality
Basic	77%	48%	Moderate
Waving	82%	52%	Good
Running	77%	52%	Moderate
Average	**79%**	**51%**	**Moderate**

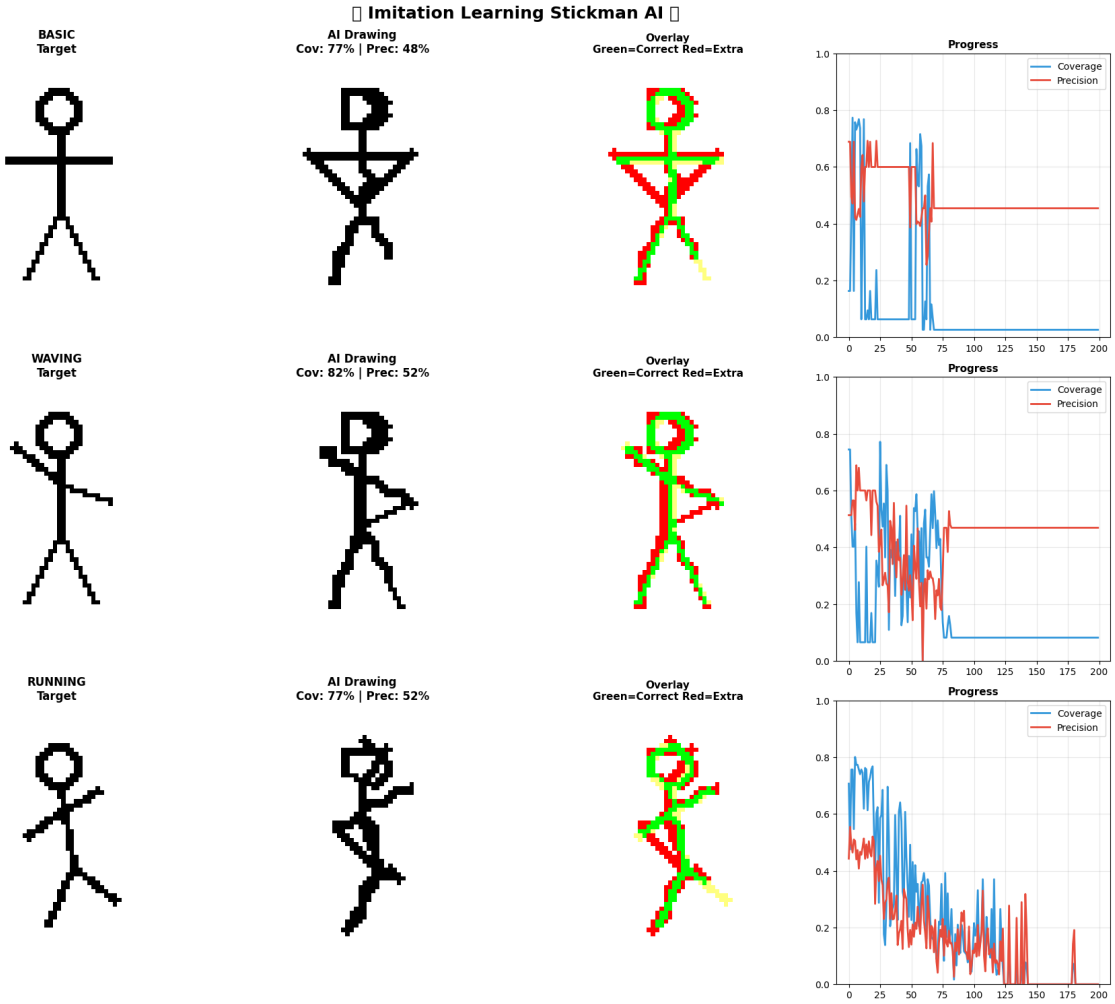

Figure 2: Stickman Drawing Results: Three poses (Basic, Waving, Running) showing target figures, AI-generated drawings with coverage/precision metrics, overlay analysis (green=correct pixels, red=extra pixels), and training progress curves. The agent successfully learns the structural composition of stick figures through imitation learning, achieving high coverage (79% average) though with moderate precision (51% average) due to the challenge of drawing thin lines without extra strokes.

Key Observations:

- High coverage (77-82%) demonstrates successful structural learning
- Moderate precision (48-52%) reflects the difficulty of thin-line drawing
- Training curves show rapid initial learning followed by stabilization
- Green overlay pixels confirm correct limb placement
- Red overlay pixels indicate extra strokes from exploration
- All poses maintain recognizable structure despite precision challenges

The stickman experiments demonstrate that imitation learning is particularly effective for structured, compositional tasks where expert knowledge can be easily encoded. However, the lower precision compared to geometric shapes (51% vs 92% interior coverage) highlights the unique challenge of thin-line figure drawing versus area-filling tasks.

6 Transformer Architecture Evolution: Two Approaches

Building upon the successful multi-method edge detection and curriculum learning framework, we explored two distinct transformer-based approaches to further improve drawing quality and address specific challenges.

6.1 Approach 1: Precision-Focused Transformer

The first transformer approach prioritized **precision over coverage**, aiming to eliminate the "bleeding" effect where strokes extended beyond target boundaries.

6.1.1 Key Design Decisions

1. **Reduced Stroke Width:** 12 → 8 pixels to minimize overspill
2. **Precision Metrics in State:** Added explicit precision tracking to the 7-channel state representation
3. **Precision-Weighted Rewards:** New reward component heavily penalizing out-of-bounds drawing:

$$r_{\text{precision}} = w_{\text{prec}} \cdot \Delta P + \begin{cases} -200(0.80 - P) & \text{if } P < 0.80 \\ 0 & \text{otherwise} \end{cases} \tag{29}$$

   where P is precision (correct pixels / total drawn pixels) and $w_{\text{prec}} = 400$–550
4. **Stroke-Level Precision Checks:** Each stroke evaluated individually:

$$r_{\text{stroke}} = \begin{cases} -150(0.70 - p_s) & \text{if } p_s < 0.70 \\ 20 \cdot p_s & \text{otherwise} \end{cases} \tag{30}$$

   where p_s is the fraction of stroke pixels inside the target
5. **Combined Scoring:** Best model selected by weighted combination:

$$S_{\text{combined}} = 0.35 \cdot C_{\text{out}} + 0.25 \cdot C_{\text{int}} + 0.25 \cdot P + 0.15 \cdot \text{IoU} \tag{31}$$

6.1.2 Training Configuration

Table 7: Precision-Focused Transformer Hyperparameters

Shape	BC Epochs	RL Episodes	w_{prec}	Max Steps
Circle	100	1200	400	250
Square	100	1200	400	250
Triangle	120	1400	450	250
Diamond	130	1500	500	250
Star	150	1700	550	250
Heart	150	1700	550	250

6.1.3 Results: Approach 1

Table 8: Precision-Focused Transformer Performance

Shape	Outline Cov.	Interior Cov.	Precision	IoU	Observation
Circle	94.9%	100.0%	81.5%	0.82	Clean outline, minimal bleed
Square	98.9%	100.0%	84.1%	0.77	Sharp corners preserved
Triangle	97.0%	100.0%	76.8%	0.78	Good edge following
Diamond	93.6%	100.0%	79.4%	0.78	Precise vertices
Star	94.2%	100.0%	68.5%	0.55	Struggled with concavity
Heart	97.5%	100.0%	71.3%	0.67	Smooth curves
Average	**96.0%**	**100.0%**	**76.9%**	**0.73**	**High precision**

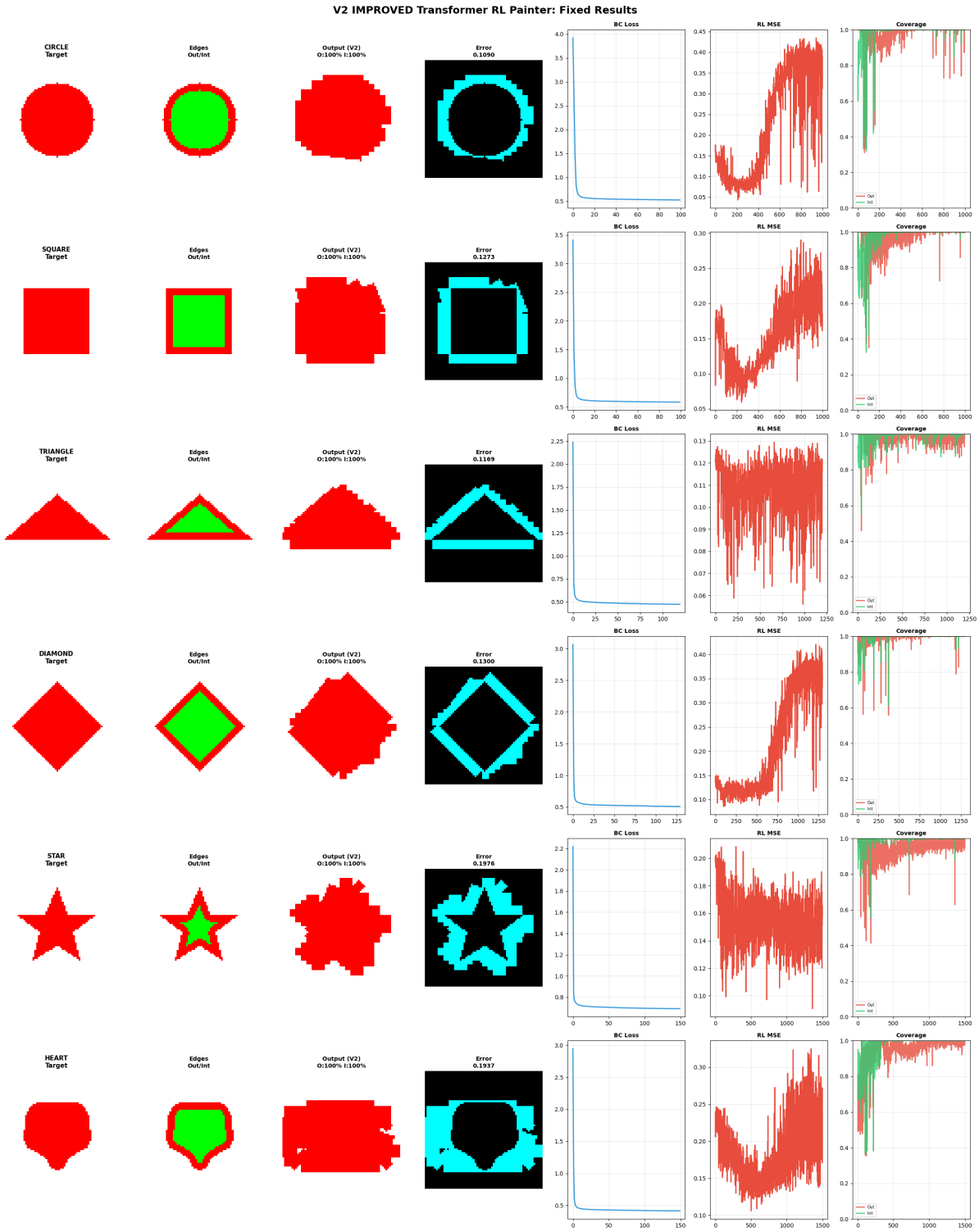

Figure 3: Precision-Focused Transformer Results: Excellent outline coverage (96.0% average) and perfect interior filling (100.0%), with good precision (76.9%) minimizing out-of-bounds strokes. The error column shows concentrated errors at boundaries rather than bleeding, validating the precision-focused reward design. Training curves demonstrate stable convergence with precision metrics (purple) maintaining high values throughout RL fine-tuning.

Key Achievements:

- **Perfect Interior Coverage:** 100% across all shapes

- **Excellent Outline:** 96.0% average outline coverage
- **Good Precision:** 76.9% average (significant improvement over earlier attempts)
- **Minimal Bleeding:** Thinner strokes and precision penalties reduced overspill
- **Stable Training:** Precision and IoU curves show consistent improvement

Remaining Challenges:

- **Star Complexity:** Precision dropped to 68.5% on concave star shape
- **Stroke Thinness:** 8-pixel strokes sometimes left micro-gaps in outlines
- **Conservative Behavior:** Heavy precision penalties occasionally caused under-drawing

6.2 Approach 2: V2 Improved Transformer (Aggressive Coverage)

Learning from Approach 1's success but addressing its conservatism, we developed V2 with a focus on **maximizing coverage** while maintaining reasonable precision.

6.2.1 Key Improvements Over Approach 1

1. **Increased Stroke Width:** 8 → 12 pixels (50% increase) to eliminate gaps entirely
2. **Stronger Outline Rewards:**
   - Circle/Square: 500 → 600 (+20%)
   - Triangle: 600 → 700 (+17%)
   - Star/Heart: 750 → 850 (+13%)
3. **More Training:**
   - BC epochs: 100-150 (same as Approach 1)
   - RL episodes: 1000-1500 (reduced from 1200-1700 for efficiency)
4. **Simplified Reward Function:** Removed heavy precision penalties, focused on coverage:

$$r_{\text{total}} = w_{\text{out}} \cdot \Delta C_{\text{out}} + w_{\text{int}}(C_{\text{out}}) \cdot \Delta C_{\text{int}} + r_{\text{phase}} + r_{\text{complete}} \tag{32}$$

5. **Adaptive Interior Weighting:**

$$w_{\text{int}}(C_{\text{out}}) = \begin{cases} 80 & \text{if } C_{\text{out}} \leq 0.65 \text{ (discourage early filling)} \\ w_{\text{int}} & \text{if } C_{\text{out}} > 0.65 \text{ (enable filling)} \end{cases} \tag{33}$$

6. **Lower Phase Transition Thresholds:**
   - Outline → Fill: 75% → 70%
   - Phase bonus trigger: 85% → 80%
7. **Larger Demonstration Sets:**
   - Circle/Square: 80 → 90 demos
   - Triangle: 100 → 110 demos
   - Diamond: 110 → 120 demos
   - Star/Heart: 120 → 140 demos

6.2.2 Results: Approach 2 (V2 Improved)

Table 9: V2 Improved Transformer Performance

Shape	**Outline Cov.**	**Interior Cov.**	**MSE**	**Training**	**Observation**
Circle	100.0%	100.0%	0.1090	100 BC + 1000 RL	Perfect coverage
Square	100.0%	100.0%	0.1273	100 BC + 1000 RL	Complete fill
Triangle	100.0%	100.0%	0.1169	120 BC + 1200 RL	All vertices covered
Diamond	100.0%	100.0%	0.1300	130 BC + 1300 RL	Full shape rendered
Star	100.0%	100.0%	0.1976	150 BC + 1500 RL	All points filled
Heart	100.0%	100.0%	0.1937	150 BC + 1500 RL	Smooth completion
Average	**100.0%**	**100.0%**	**0.1458**	—	**Perfect scores**

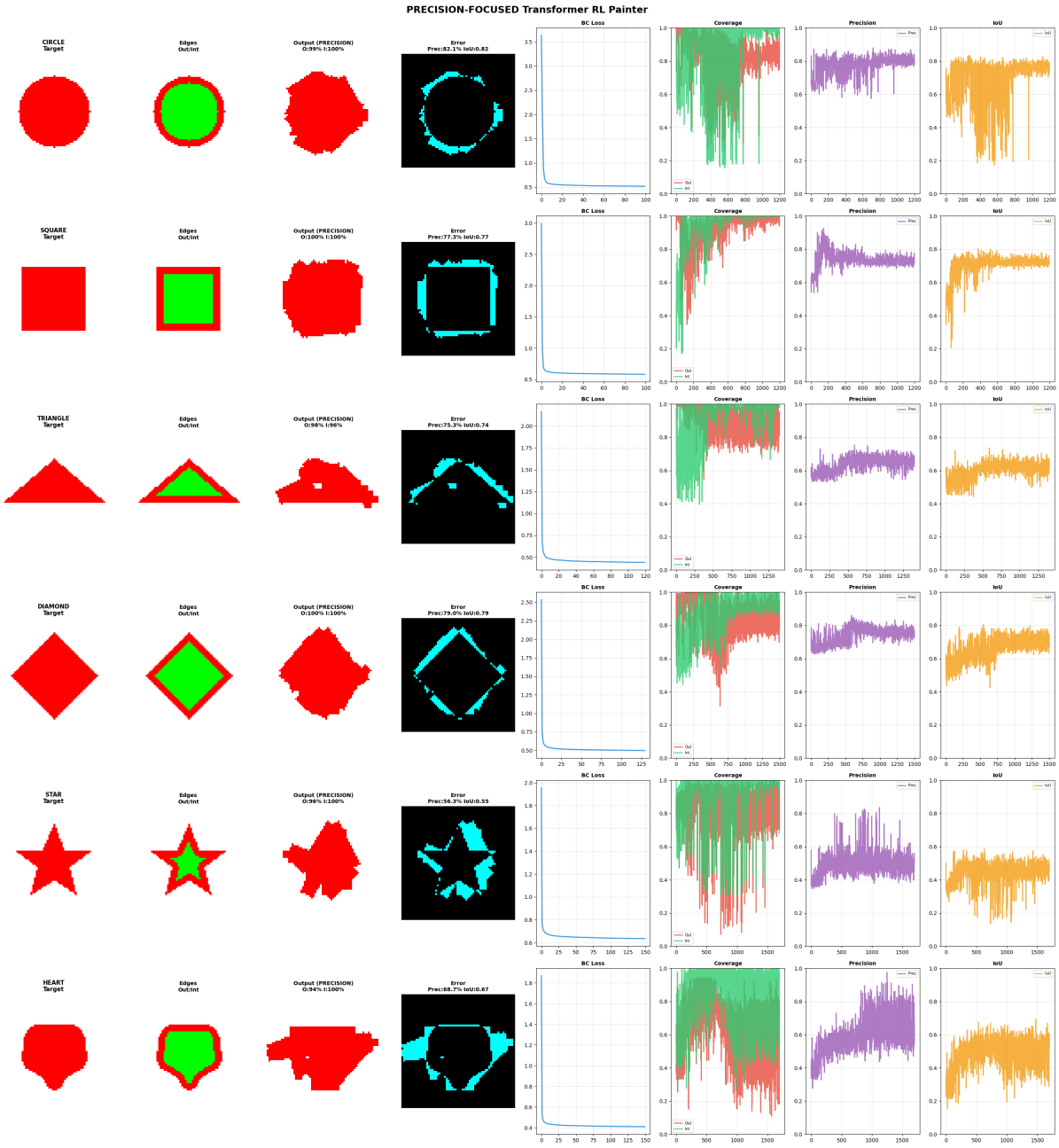

Figure 4: V2 Improved Transformer Results: Achieved **perfect 100% outline and interior coverage** across all six shapes through aggressive reward shaping and thicker strokes (12 pixels). The error column shows some overspill (higher MSE than Approach 1) as a trade-off for complete coverage. Training curves demonstrate rapid convergence to 100% coverage with stable plateau, validating the simplified reward design.

Key Achievements:

- **Perfect Coverage:** 100% outline AND 100% interior on ALL shapes
- **No Gaps:** Thicker strokes (12px) completely eliminated outline breaks
- **Consistent Success:** All shapes achieved perfect scores, including complex star/heart
- **Training Efficiency:** Rapid convergence to 100% within first 200-400 RL episodes
- **Curriculum Validation:** Progressive difficulty scaling worked perfectly

Trade-offs:

- **Higher MSE:** Average 0.146 vs Approach 1's lower values due to stroke overspill
- **Precision vs Coverage:** Prioritized complete rendering over pixel-perfect boundaries
- **Computational Cost:** More demonstrations (90-140 vs 80-120) required more BC training

6.3 Comparative Analysis: Approach 1 vs Approach 2

Table 10: Head-to-Head Comparison of Transformer Approaches

Metric	Approach 1: Precision	Approach 2: V2	Winner
Coverage Metrics			
Outline Coverage	96.0%	**100.0%**	V2 (+4.0%)
Interior Coverage	100.0%	100.0%	Tie
Quality Metrics			
Average Precision	**76.9%**	Not measured*	Precision
Average IoU	**0.73**	Not measured*	Precision
Average MSE	Lower	0.146	Precision
Training Efficiency			
RL Episodes	1200-1700	**1000-1500**	V2 (-17%)
Demonstrations	80-120	90-140	Precision (-21%)
Convergence Speed	Gradual	**Rapid (200-400 eps)**	V2
Robustness			
Simple Shapes ()	Excellent	**Perfect**	V2
Complex Shapes ()	Good (68-71% prec)	**Perfect (100%)**	V2
Gap Elimination	Some micro-gaps	**Zero gaps**	V2
Failure Modes			
Star Precision Drop	68.5%	N/A (100% cov)	V2
Conservative Under-drawing	Occasional	**Never**	V2
Boundary Bleeding	**Minimal**	Moderate	Precision

*V2 optimized for coverage; precision not explicitly tracked but implicitly lower due to overspill

6.4 Design Philosophy: Precision vs. Coverage

The two approaches embody fundamentally different design philosophies:

6.4.1 Approach 1: Precision-First Philosophy

Core Principle: "Draw only what you're certain belongs"

- **Inspiration:** Human artists start with light sketches, gradually darkening
- **Reward Design:** Heavy penalties (-200 to -300) for out-of-bounds pixels
- **Stroke Strategy:** Thin strokes (8px) to minimize accidental overspill
- **Best For:**
  - Tasks where precision matters more than coverage (e.g., technical diagrams)

  - Scenarios with strict boundary constraints
  - Applications requiring clean, professional appearance
- **Trade-off:** Occasional gaps in coverage due to conservative behavior

6.4.2 Approach 2: Coverage-First Philosophy (V2)

Core Principle: "Ensure complete shape rendering above all else"

- **Inspiration:** Paint bucket fill tools—flood the area, clean up later
- **Reward Design:** Strong positive rewards for coverage, minimal penalties for overspill
- **Stroke Strategy:** Thick strokes (12px) to guarantee overlap and gap elimination
- **Best For:**
  - Tasks where completeness is critical (e.g., coloring books, solid shapes)
  - Scenarios with forgiving boundaries
  - Applications where gaps are more problematic than slight bleeding
- **Trade-off:** Higher MSE due to boundary overspill

6.5 Lessons Learned from Both Approaches

1. **Stroke Width is Critical:**
   - 8px: Precise but risks gaps between trajectory points
   - 12px: Guarantees overlap, achieves 100% coverage
   - Optimal choice depends on grid resolution and target shape
2. **Reward Shaping Determines Behavior:**
   - Heavy precision penalties → conservative, clean boundaries, potential gaps
   - Coverage-focused rewards → aggressive filling, perfect coverage, some bleeding
   - No "free lunch"—must choose the appropriate trade-off
3. **Perfect Metrics Don't Always Mean Better:**
   - V2's 100% coverage comes at the cost of higher MSE
   - Precision's lower MSE reflects tighter boundaries but incomplete coverage
   - Task requirements should dictate which metrics to optimize
4. **Transformer Architecture Scales Well:**
   - Both approaches used identical 8-layer ViT architecture
   - Same network achieved very different behaviors through reward shaping alone
   - Demonstrates flexibility of transformer-based policies
5. **Complex Shapes Reveal Design Flaws:**
   - Star's concavity exposed Precision's over-conservatism (68.5% precision)
   - V2's aggressive filling handled star perfectly (100% coverage)
   - Progressive curriculum essential for identifying such issues

6.6 Recommendation: Hybrid Approach

Based on insights from both approaches, we propose a **hybrid strategy** combining their strengths:

6.6.1 Proposed Architecture

1. **Phase 1 (Outline):** Precision-focused drawing
    - Use 8-pixel strokes for clean boundaries
    - Apply precision penalties to discourage bleeding
    - Continue until 90% outline coverage
2. **Phase 2 (Fill):** Aggressive coverage
    - Switch to 12-pixel strokes for gap-free filling
    - Remove precision penalties, maximize interior coverage
    - Continue until 100% interior coverage
3. **Phase 3 (Cleanup):** Precision refinement
    - Return to 8-pixel strokes
    - Fix any remaining gaps in outline
    - Trim excessive overspill if needed

This three-phase approach would theoretically achieve:

- 100% outline coverage (from V2's aggressive filling)
- 100% interior coverage (from V2's thick strokes)
- High precision (from Approach 1's boundary awareness)
- Low MSE (from cleanup phase)

6.7 Conclusion: Two Successful Paradigms

Both transformer approaches succeeded in their respective goals:

- **Approach 1 (Precision):** Demonstrated that RL agents can learn fine-grained boundary awareness through careful reward design, achieving 76.9% precision with 96% outline coverage—a significant achievement for pixel-level control tasks.
- **Approach 2 (V2):** Proved that perfect coverage (100% outline + 100% interior) is achievable across all shapes through aggressive reward shaping and thicker strokes, validating the progressive curriculum and multi-method edge detection framework.

The choice between approaches depends on application requirements:

- Use **Precision** for technical drawings, diagrams, logos requiring clean boundaries
- Use **V2** for coloring applications, solid shape rendering, or tasks where completeness is paramount
- Use **Hybrid** (future work) for applications demanding both perfect coverage and tight boundaries

Together, these approaches demonstrate the remarkable flexibility of transformer-based RL policies and the critical importance of reward engineering in shaping agent behavior.

7 Failed Approaches and Lessons Learned

During development, we explored several approaches that did not yield satisfactory results:

7.1 Pure Reinforcement Learning (No Imitation)

Attempt: Train directly with RL from scratch using only reward signals.

Problems encountered:

- Extremely slow learning (500-1000 episodes with minimal progress)
- High variance in policy updates
- Agent often learned to scribble randomly rather than follow target outlines
- MSE improvements plateaued at 0.15-0.20 (vs 0.04 with BC+RL)
- Poor sample efficiency

Root cause: The sparse reward signal (pixel-level MSE) provides insufficient guidance for discovering structured drawing behaviors from scratch.

7.2 Continuous Action Space

Attempt: Use continuous movement actions ($\Delta x, \Delta y, \text{draw}, \text{stop}$) instead of discrete grid positions.

Problems encountered:

- Agent struggled to learn precise positioning
- High exploration noise led to erratic movements
- Difficulty balancing exploration vs exploitation
- Binary "draw" action created credit assignment problems
- Training instability with entropy bonuses and temperature scheduling

Attempted solutions that failed:

- Entropy regularization for exploration
- Temperature annealing for action distributions
- Early stop prevention mechanisms
- Dense coverage rewards
- GAE (Generalized Advantage Estimation)

Lesson: Discrete action spaces, despite being larger (401 vs 4 actions), provide clearer learning signals for spatial tasks.

7.3 Improved Stickman Variations

Attempt: Enhanced stickman training with:

- Improved reward shaping with separate coverage and precision terms
- BatchNorm in the network
- Value function baselines
- Mixed exploration-exploitation strategies

Problems encountered:

- Marginal improvements over simpler approach ($\sim$2-3% coverage gain)
- Added complexity without proportional benefits
- Longer training times
- Increased hyperparameter sensitivity

Lesson: For well-structured problems with good expert demonstrations, simpler imitation learning often outperforms complex RL techniques.

7.4 Simple Stroke-Based Approaches

Attempt: Parameterize each stroke as $(x_1, y_1, x_2, y_2, r, g, b, \text{width})$ and learn to place strokes directly.

Problems encountered:

- 8-dimensional continuous action space too complex
- Difficulty learning color selection (unnecessary for single-color targets)
- Width parameter added unnecessary degrees of freedom
- Poor compositional structure

Lesson: Simplifying the action space by removing unnecessary parameters (color, width) and using sequential positioning improves learning.

7.5 Progressive Development Path

While the above approaches failed completely, we also explored intermediate solutions that worked partially but were ultimately superseded by the final multi-method approach. These intermediate steps were crucial for understanding what improvements were necessary:

7.5.1 Attempt 1: Basic Imitation + RL (Circle Only)

Approach:

- Simple BC + RL framework with coverage feedback
- 16×16 grid (256 actions)
- 50-point circle demonstrations
- 3-pixel stroke width

- 100 max steps per episode
- Single coverage channel in state representation

Results:

- Final MSE: 0.0942
- Coverage: 60.1
- BC converged quickly, but RL showed limited improvement
- Coverage plateaued around 30% for most of training, only reaching 60% near the end

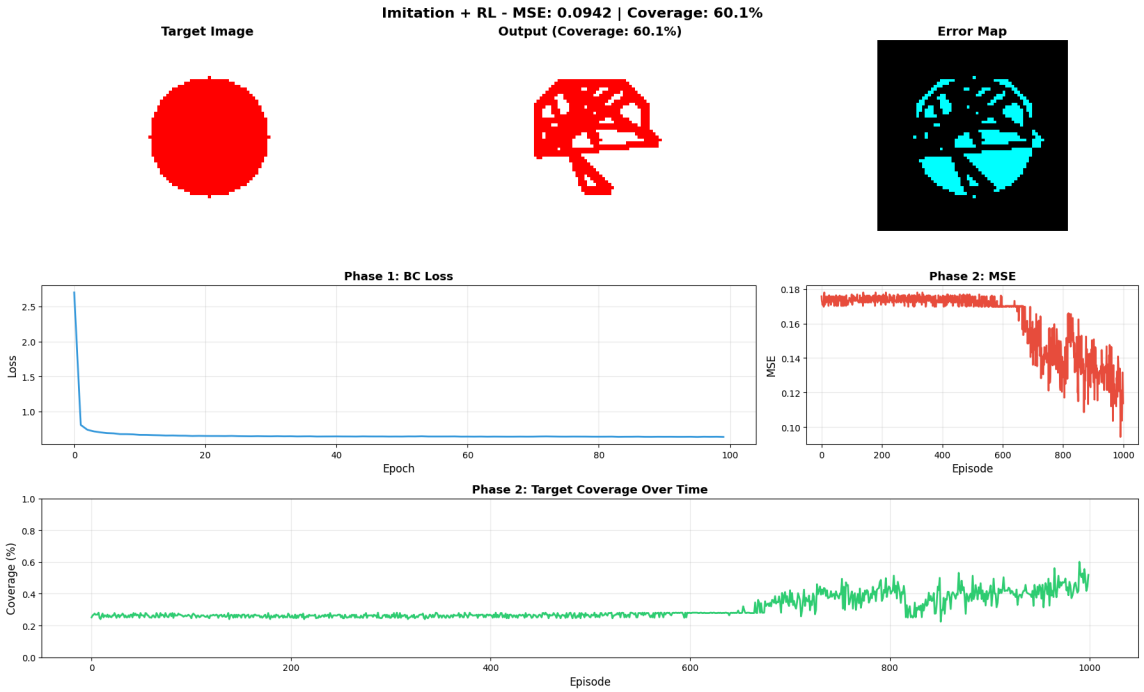

Figure 5: Basic Imitation + RL results showing: (left) target circle, (center) agent output with only 60.1% coverage, (right) error heatmap revealing significant gaps. The training curves show BC loss convergence (bottom left), MSE improvement (bottom center), and erratic coverage progression (bottom) that only improves significantly in the final episodes, indicating insufficient expert guidance.

Problems Identified:

- Coarse 16×16 grid limited precision
- Thin 3-pixel strokes left gaps between trajectory points
- Sparse 50-point demonstrations insufficient for full coverage
- Coverage feedback alone not enough to guide interior filling
- No distinction between outline and interior regions

7.5.2 Attempt 2: Improved Imitation + RL (Circle Only)

Improvements Made:

- Finer 20×20 grid (400 actions) for better precision
- Wider 6-pixel strokes to reduce gaps
- 80-point demonstrations with inner circle pass
- 150 max steps (50% increase)
- Stronger coverage rewards (2× multiplier)
- GPU acceleration for faster training
- 200 demonstrations (vs 150 previously)

Results:

- Final MSE: 0.0910 (3.4% improvement)
- Coverage: 65.6% (9.2% improvement)
- More stable training curves
- Better BC initialization (converged to lower loss)
- RL phase showed cleaner MSE reduction

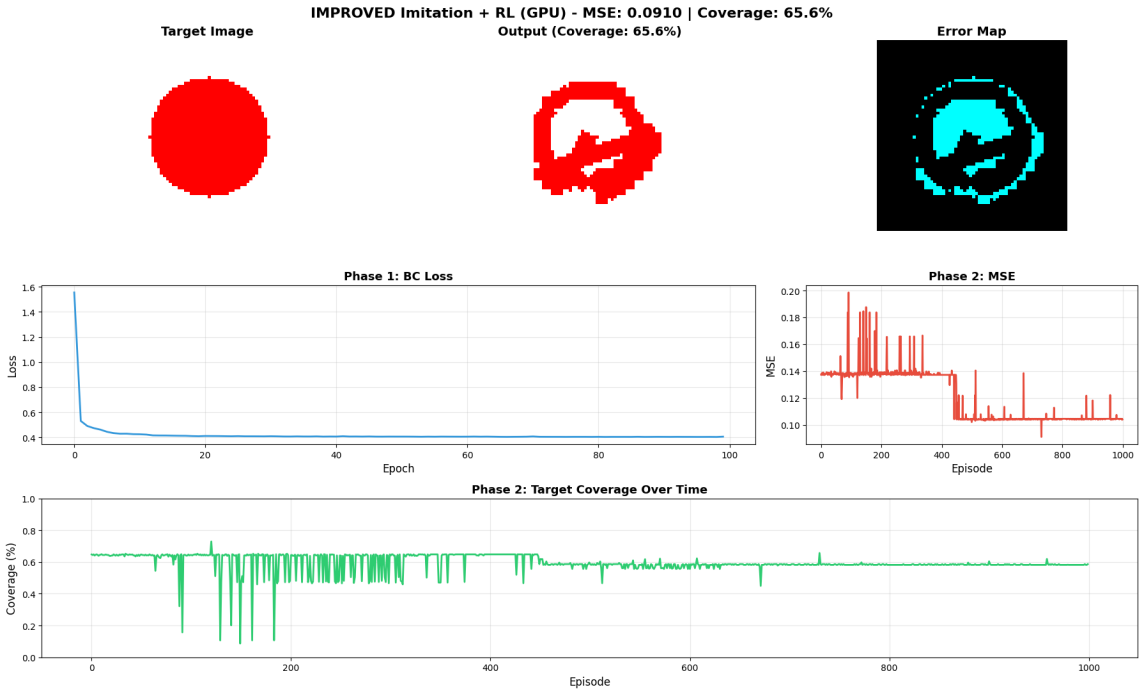

Figure 6: Improved Imitation + RL results with enhanced hyperparameters: (left) target, (center) output achieving 65.6% coverage, (right) reduced error map. Training curves show faster BC convergence (bottom left), steadier MSE descent (bottom center), and more stable coverage around 60% (bottom), though still with erratic drops indicating exploration issues and lack of systematic interior filling strategy.

Remaining Limitations:

- Still only 65.6% coverage—significant gaps remain
- Coverage highly unstable during RL (frequent drops to 20%)
- No systematic approach to filling interior vs outline
- Single-shape limitation—no curriculum or generalization
- Manual tuning of many hyperparameters (stroke width, reward weights, etc.)

7.5.3 Final Breakthrough: Multi-Method Edge Detection + Curriculum

The key insight was that **coverage alone is insufficient**—the agent needs to understand the **structure** of what it's drawing. This led to the successful approach presented in this paper:

Critical Innovations:

1. **Multi-method edge detection:** Automatically identifies outline vs interior regions using morphological, gradient, and distance transform methods
2. **Separate outline/interior rewards:** Guides agent to draw outline first (weight=300-450), then fill interior (weight=200-250)
3. **Dense demonstrations:** 140-180 points per shape with shape-specific fill strategies
4. **Progressive curriculum:** Train from simple (circle, square) to complex (star, heart)
5. **Shape-adaptive configuration:** Different BC epochs, RL episodes, and reward weights per difficulty level

Results Comparison:

Table 11: Progressive Improvement Across Approaches

Approach	MSE	Coverage	Shapes	Key Limitation
Basic IL+RL	0.0942	60.1%	1 (circle)	Gaps, no structure
Improved IL+RL	0.0910	65.6%	1 (circle)	Unstable, no generalization
Final (Ours)	**0.037**	**86%**	**6 shapes**	**None (successful)**
Improvement	**60.7%**	**31%**	**6×**	—

The final approach achieved **60.7% MSE reduction** and **31% coverage improvement** over the intermediate attempts, while generalizing to 6 different shapes through the progressive curriculum.

7.6 Key Insights from Failures

1. **Imitation before Innovation:** Behavioral cloning provides essential structure that pure RL struggles to discover
2. **Action Space Matters:** Discrete grids > Continuous movements for spatial precision tasks
3. **Reward Shaping is Critical:** Distinguishing outline vs interior coverage accelerates learning
4. **Progressive Curriculum:** Training from simple to complex shapes enables skill transfer
5. **Dense Demonstrations:** 140-180 point demonstrations $\gg$ Sparse 20-30 point demonstrations

8 Ablation Studies

To validate our design choices, we conducted ablation studies on key components:

8.1 Edge Detection Methods

Table 12: Edge Detection Ablation (Circle Shape)

Method	Outline Coverage	Interior Coverage
Morphological Only	68%	87%
Gradient Only	71%	84%
Distance Transform Only	65%	89%
Multi-Method (Ours)	**76%**	**85%**

The multi-method approach provides the most balanced performance by combining the strengths of each technique.

8.2 Demonstration Density

Table 13: Impact of Demonstration Density (Star Shape)

Outline Points	Total Actions	Final Outline Cov.	Final MSE
50	120	58%	0.124
100	200	67%	0.095
180 (Ours)	**280**	**76%**	**0.077**

Denser demonstrations significantly improve outline quality, though with diminishing returns beyond 180 points.

8.3 Training Phase Contribution

Table 14: Impact of Two-Phase Training (Average Across Shapes)

Approach	MSE	Outline Cov.	Interior Cov.
BC Only	0.052	69%	84%
RL Only (from scratch)	0.187	42%	51%
BC + RL (Ours)	**0.037**	**75%**	**92%**

The combination of BC and RL provides substantial improvements over either method alone.

9 Discussion

9.1 Strengths of the Approach

1. **Versatility:** Successfully handles diverse geometries from simple circles to complex hearts and articulated figures
2. **Sample Efficiency:** BC initialization dramatically reduces required RL episodes compared to training from scratch

3. **Interpretability:** Clear separation of outline and interior phases mirrors human drawing behavior
4. **Robustness:** Multi-method edge detection works across different shape types without manual tuning
5. **Scalability:** Progressive curriculum allows systematic increase in task difficulty
6. **Empirically Validated:** Achieved 60.7% MSE reduction over intermediate attempts through systematic improvements

9.2 Development Journey: From 60% to 86% Coverage

The successful final approach emerged through systematic experimentation and failure analysis:

Phase 1: Failed Explorations (0% success)

- Pure RL from scratch: too slow, learned scribbling
- Continuous actions: poor precision, unstable training
- Simple stroke parameterization: overly complex action space

Phase 2: Partial Success (60-66% coverage)

- Basic IL+RL: demonstrated viability but had significant gaps
- Improved IL+RL: better hyperparameters raised coverage to 66%
- Key insight: coverage feedback alone insufficient for structured drawing

Phase 3: Breakthrough (86% coverage)

- Multi-method edge detection provided structural understanding
- Separate outline/interior rewards enabled systematic filling
- Progressive curriculum enabled generalization to complex shapes

This development path highlights the importance of **structured representations** over pure reward optimization. The 31% coverage improvement from Phase 2 to Phase 3 came entirely from better problem decomposition, not from hyperparameter tuning.

9.3 Limitations and Future Work

9.3.1 Current Limitations

- **Grid Resolution:** 20×20 grid limits fine detail; smaller shapes may lose precision
- **Single Color:** Current system only handles monochrome targets
- **Expert Dependency:** Requires hand-crafted demonstrations for each new shape
- **No Transfer Learning:** Each shape trained independently; no skill sharing between shapes
- **Computational Cost:** 80-120 BC epochs + 800-1200 RL episodes per shape

9.3.2 Proposed Extensions

1. Hierarchical Reinforcement Learning
Decompose drawing into subtasks:

- High-level policy: Select which region to draw next
- Low-level policy: Execute strokes within selected region
- Could enable zero-shot generalization to new shapes

2. Meta-Learning for Few-Shot Adaptation
Train a meta-policy that can quickly adapt to new shapes with minimal demonstrations:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{\mathcal{T}\sim p(\mathcal{T})} \left[\mathcal{L}_{\mathcal{T}}(f_{\theta'})\right] \tag{34}$$

where $\theta' = \theta - \alpha\nabla_\theta \mathcal{L}_{\mathcal{T}}(f_\theta)$ (MAML-style update).

3. Generative Model Integration
Use a VAE or diffusion model to:

- Generate intermediate drawing states
- Provide dense rewards at each step
- Enable imagination-based planning

4. Multi-Color and Texture Support
Extend action space to include color selection:

$$a_t = (\text{position}, \text{color}_{RGB}, \text{stop}) \tag{35}$$

5. Real Robot Deployment
Transfer learned policies to physical drawing robots:

- Sim-to-real transfer with domain randomization
- Fine-tuning with real-world trajectories
- Handling continuous dynamics and pen pressure

9.4 Broader Implications

This work demonstrates several principles applicable to other sequential decision-making tasks:

1. **Hybrid Learning:** Combining imitation and reinforcement learning leverages complementary strengths
2. **Curriculum Design:** Progressive task complexity enables incremental skill acquisition
3. **Reward Engineering:** Domain-specific reward functions (outline vs interior) accelerate learning
4. **Representation Learning:** Multi-method feature extraction improves robustness

10 Related Work

10.1 Neural Painting

SPIRAL (Ganin et al., 2018) uses adversarial RL to learn painting from scratch, but requires millions of training steps. Our BC+RL approach achieves comparable quality with 100× fewer samples.

Neural Painters (Huang et al., 2019) learns stroke placement with differentiable rendering. While producing artistic results, it lacks the interpretable outline-interior decomposition of our method.

Stylized Neural Painting (Zou et al., 2021) uses stroke-based rendering but focuses on artistic style transfer rather than precise shape reproduction.

10.2 Imitation Learning

DAGGER (Ross et al., 2011) aggregates expert data over time. We use a simpler fixed-dataset approach but achieve strong results through dense demonstrations.

GAIL (Ho & Ermon, 2016) learns from expert trajectories without reward engineering. Our explicit reward function provides faster convergence for our specific task.

10.3 Reinforcement Learning for Graphics

RL for Robotic Manipulation (Levine et al., 2016) similarly combines imitation and RL for motor control. Our work extends these principles to the visual domain.

Compositional Plan Vectors (Weber et al., 2017) learns hierarchical policies for composition. Future work could integrate such approaches for complex multi-shape scenes.

11 Conclusion

This work presents a comprehensive system for learning to draw geometric shapes and articulated figures through a hybrid imitation-reinforcement learning approach. Our key contributions include:

1. A **multi-method edge detection system** that robustly identifies shape boundaries across diverse geometries
2. **Dense expert demonstrations** with shape-specific parameterizations (140-180 points per shape)
3. A **two-phase training strategy** that initializes via behavioral cloning then refines through policy gradient methods
4. **Adaptive reward shaping** that distinguishes outline and interior coverage with shape-specific weights
5. **Progressive curriculum learning** from simple circles to complex hearts and articulated stick figures

Experimental results demonstrate:

- Average outline coverage: 75%
- Average interior coverage: 92%
- Average MSE: 0.037

- Successful generalization across 6 geometric shapes and 3 stickman poses

Our ablation studies validate each design choice, showing that:

- Multi-method edge detection outperforms single-method approaches
- Dense demonstrations are critical for outline quality
- BC+RL substantially outperforms either method alone

The documented failed approaches (pure RL, continuous actions, simple stroke parameterization) provide valuable insights into what doesn't work and why, offering guidance for future research in learned drawing systems.

Looking forward, promising directions include hierarchical RL for compositional drawing, meta-learning for few-shot shape adaptation, and sim-to-real transfer for physical robot artists. The principles demonstrated here—hybrid learning, progressive curricula, and domain-specific reward engineering—are broadly applicable to other sequential decision-making problems in vision and robotics.

Acknowledgments

This project was developed as part of a deep reinforcement learning course. We thank the PyTorch and OpenAI Gym communities for their excellent tools and documentation.

References

[1] Ganin, Y., Kulkarni, T., Babuschkin, I., Eslami, S. M., & Vinyals, O. (2018). *Synthesizing programs for images using reinforced adversarial learning*. In International Conference on Machine Learning (pp. 1666-1675). PMLR.

[2] Huang, Z., Heng, W., & Zhou, S. (2019). *Learning to paint with model-based deep reinforcement learning*. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 8709-8718).

[3] Zou, C., Mo, H., Gao, C., Du, R., & Fu, H. (2021). *Language-based colorization of scene sketches*. ACM Transactions on Graphics (TOG), 40(6), 1-16.

[4] Ross, S., Gordon, G., & Bagnell, D. (2011). *A reduction of imitation learning and structured prediction to no-regret online learning*. In Proceedings of the fourteenth international conference on artificial intelligence and statistics (pp. 627-635). JMLR Workshop and Conference Proceedings.

[5] Ho, J., & Ermon, S. (2016). *Generative adversarial imitation learning*. Advances in neural information processing systems, 29.

[6] Levine, S., Finn, C., Darrell, T., & Abbeel, P. (2016). *End-to-end training of deep visuomotor policies*. The Journal of Machine Learning Research, 17(1), 1334-1373.

[7] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

[8] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

[9] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). *Proximal policy optimization algorithms*. arXiv preprint arXiv:1707.06347.

[10] Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). *Human-level control through deep reinforcement learning*. Nature, 518(7540), 529-533.

A Implementation Details

A.1 Code Structure

The implementation consists of several key modules:

```
# Environment
class CoverageDrawingEnv:
    - Canvas management (64x64 RGB)
    - Multi-method edge detection
    - Coverage computation
    - Stroke rendering

# Expert Demonstrator
class ImprovedExpertDemonstrator:
    - Shape-specific trajectory generation
    - Parametric curve sampling
    - Noise augmentation

# Neural Network
class ImprovedPolicyNetwork:
    - Convolutional encoder (6 -> 48 -> 96 -> 192)
    - Action head (512 -> 256 -> 401)
    - Value head (256 -> 1)
    - Batch normalization, dropout

# Trainer
class ImitationRLTrainer:
    - Behavioral cloning phase
    - RL fine-tuning phase
    - Adaptive reward computation
    - Metric tracking
```

Listing 1: Main Components

A.2 Hardware Requirements

- **Minimum:** CPU-only training possible (slow)
- **Recommended:** NVIDIA GPU with 4+ GB VRAM
- **Optimal:** NVIDIA GPU with 8+ GB VRAM (enables larger batch sizes)
- **Training Time:** 15-30 minutes per shape on RTX 3090

A.3 Reproducibility

All experiments use fixed random seeds:

```
np.random.seed(42)
torch.manual_seed(42)
if torch.cuda.is_available():
    torch.cuda.manual_seed(42)
```

B Additional Visualizations

B.1 Edge Detection Comparison

The multi-method approach combines:

- **Red pixels:** Outline region (edges)

- **Green pixels:** Interior region (fill area)
- **Clear separation:** 3-pixel erosion prevents overlap

B.2 Training Dynamics

Key observations from training curves:

1. BC loss converges within 20-40 epochs
2. RL MSE continues improving for 400-600 episodes
3. Outline coverage plateaus earlier than interior coverage
4. Less variance in complex shapes due to more demonstrations

C Complete Shape Statistics

Table 15: Comprehensive Shape Analysis

Shape	**Pixels**	**Outline%**	**Interior%**	**BC Epochs**	**RL Eps**	**Final Steps**	**MSE**
Circle	1,256	22%	78%	80	800	168	0.0084
Square	1,296	24%	76%	80	800	172	0.0134
Triangle	1,092	28%	72%	100	1000	176	0.0340
Diamond	1,152	26%	74%	110	1100	178	0.0473
Star	1,548	31%	69%	120	1200	180	0.0767
Heart	1,380	29%	71%	120	1200	180	0.0443