Adaptive WM

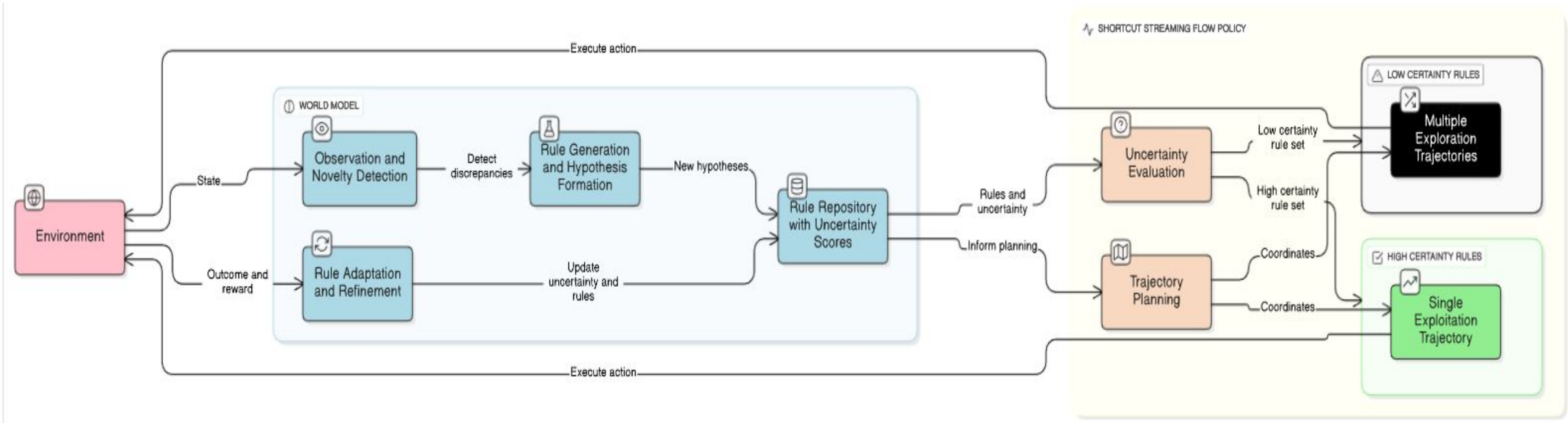
Pranav Lakshmanan

Overall Architecture

RQ- Can we have adaptive world model that can observe , reason and adapt it understanding of the world as the agent explores by modifying its own rules?

Proposed architecture -

World Model (WM) = The scientist (generates hypotheses, maintains rules)
Shortcut Streaming Flow Policy = The experimenter (efficiently tests hypotheses by jumping through state-action space/state space)

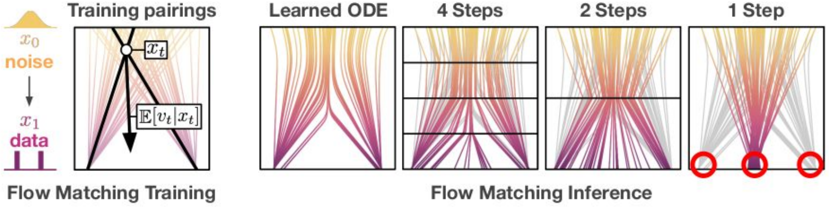


SHORTCUT STREAMING FLOW POLICY
Execute action
WORLD MODEL
Environment
State
Observation and Novelty Detection
Detect discrepancies
Rule Generation and Hypothesis Formation
New hypotheses
Rule Repository with Uncertainty Scores
Outcome and reward
Rule Adaptation and Refinement
Update uncertainty and rules
Rules and uncertainty
Inform planning
Uncertainty Evaluation
Trajectory Planning
Low certainty rule set
High certainty rule set
Coordinates
Coordinates
LOW CERTAINTY RULES
Multiple Exploration Trajectories
HIGH CERTAINTY RULES
Single Exploitation Trajectory
Execute action

What are shortcuts

RQ- How can we make diffusion models generate in ONE step instead of 128 steps, without sacrificing quality?

$$x_t = (1-t)\,x_0 + t\,x_1 \qquad \text{and} \qquad v_t = x_1 - x_0.$$



$$\bar{v}_\theta(x_t, t) \approx \mathbb{E}_{x_0, x_1 \sim D}\left[v_t \mid x_t\right] \qquad \mathcal{L}^{\mathrm{F}}(\theta) = \mathbb{E}_{x_0, x_1 \sim D}\left[||\bar{v}_\theta(x_t, t) - (x_1 - x_0)||^2\right]$$

Shortcuts

Condition the network on BOTH:

1. **Current noise level (t):** Where am I in the denoising process?
2. **Desired step size (d):** How big of a jump am I taking?

Self Consistency property of shortcuts

$$x'_{t+d} = x_t + s(x_t, t, d)\, d.$$

$$s(x_t, t, 2d) = s(x_t, t, d)/2 + s(x'_{t+d}, t+d, d)/2$$

$$\mathcal{L}^{\mathrm{S}}(\theta) = E_{x_0 \sim \mathcal{N},\, x_1 \sim D,\, (t,d) \sim p(t,d)} \Big[\underbrace{\| s_\theta(x_t, t, 0) - (x_1 - x_0) \|^2}_{\text{Flow-Matching}} + \underbrace{\| s_\theta(x_t, t, 2d) - s_{\text{target}} \|^2}_{\text{Self-Consistency}} \Big],$$

$$\text{where} \quad s_{\text{target}} = s_\theta(x_t, t, d)/2 + s_\theta(x'_{t+d}, t, d)/2 \quad \text{and} \quad x'_{t+d} = x_t + s_\theta(x_t, t, d) d.$$

Example and Implications of shortcut models

1. Ask model: "From $x=2$, take step $d=0.25$" → model says "go +3" → land at $x=2.75$
2. Ask model: "From $x=2.75$, take step $d=0.25$" → model says "go +3.2" → land at $x=5.95$
3. Target for $d=0.5$: average = $(3 + 3.2)/2 = 3.1$
4. Train: $s(x=2, d=0.5)$ should output 3.1

By conditioning on d, one model learns to handle ANY inference budget (1 step, 10 steps, 100 steps) at test time. You choose speed vs quality AFTER training, not before.

For future state prediction: Same model can predict 1-step ahead (fast, less accurate) or 50-steps ahead (slow, more accurate) depending on the WM needs .

Shortcuts for future state prediction

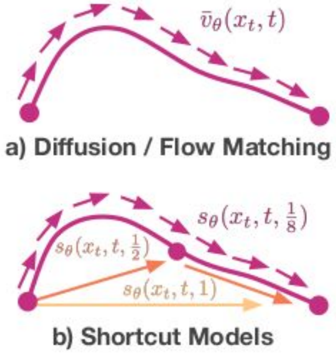
Velocity field for state space -

$$v_\theta : S \times A^k \times \mathbb{R}_+ \to T_S$$

Rate change of state -

$$\frac{ds}{dt} = v_\theta(s(t), \mathbf{a}, t)$$

- A state s (where you are now)
- An action sequence $\mathbf{a}$ (what you plan to do)
- Time t (when you're doing it)



The exact solution - (but we need to know s(τ) at every intermediate time τ)

$$s_{t+d} = s_t + \int_t^{t+d} v_\theta(s(\tau), \mathbf{a}, \tau)\, d\tau$$

Shortcut approximation

Approximation -

$$s_\theta(s_t, \mathbf{a}, t, d) \approx s_t + v_\theta(s_t, \mathbf{a}, t) \cdot d$$

Self consistency in state space -

$$s_\theta(s_t, \mathbf{a}, t, 2d) = s_\theta(s_t, \mathbf{a}, t, d) + s_\theta(s_{t+d}, \mathbf{a}', t + d, d)$$

Velocity matching -

$$\mathcal{L}_v = \|v_\theta(s_t, \mathbf{a}, t) - \dot{s}_{\text{true}}(s_t, \mathbf{a}, t)\|^2$$

Self consistency loss -

$$\mathcal{L}_{\text{flow}} = \|s_\theta(s_t, \mathbf{a}, t, 2d) - [s_\theta(s_t, \mathbf{a}, t, d) + s_\theta(s_{t+d}, \mathbf{a}', t + d, d)]\|^2$$

Inputs needed

- Any obvious oversight in this approach (in terms of math or concepts)
- Suggestions for simulators/worlds that can validate this approach quickly?
- Any particular evals to run ?
- Where do you think more mathematical rigour is needed?
- Baselines to compare against?