AdaFlow: Imitation Learning with Variance-Adaptive Flow-Based Policies

Xixi Hu, Bo Liu, Xingchao Liu, Qiang Liu @ UT Austin

What This is About:

We show a simple way to train a robot's action-generating model with rectified flow (flow matching), so it can **produce actions in just one step** (almost)—no extra complicated training steps, such as distillation needed.

How the Method Works:

- We use rectified flow (flow matching) to train the robot's action model
- The method learns a flow (or diffusion) policy with expert demonstrations
- During training, the model predicts action and **state variance**.
- State variance shows if the action deterministic or uncertain.
- For low-variance states, it generates actions in one step, like Behavior Cloning (BC)
- For high-variance states, it takes more steps to ensure accuracy.
- **Key advantage:** fast and efficient, with no extra training steps like distillation.

How to Apply the Method to Your Robot Policy:

- Make your model predict a scalar value (variance) in addition to the actions.
- Train your model using the following loss function:

$$\min_{\phi} \mathbb{E}\left[\int_0^1 \frac{\|\boldsymbol{a}-\boldsymbol{x}_0-v_\theta(\boldsymbol{x}_t,t|\boldsymbol{s})\|^2}{2\sigma_\phi^2(\boldsymbol{x}_t,t|\boldsymbol{s})}+\log\sigma_\phi^2(\boldsymbol{x}_t,t|\boldsymbol{s})\mathrm{d}t\right].$$

a GT action x_0 Random Gaussian noise v_θ Policy model σ_ϕ Predicted variance

- Use the following algorithm for sampling actions:

Algorithm 1 AdaFlow: Execution

1: **Input:** Current state $\boldsymbol{s}$, minimal step size $\epsilon_{\min}$, error threshold η, pre-trained networks v_θ and σ_ϕ.
2: Initialize $\boldsymbol{z}_0 \sim \mathcal{N}(0, I)$, $t = 0$.
3: **while** $t < 1$ **do**
4: Compute step size

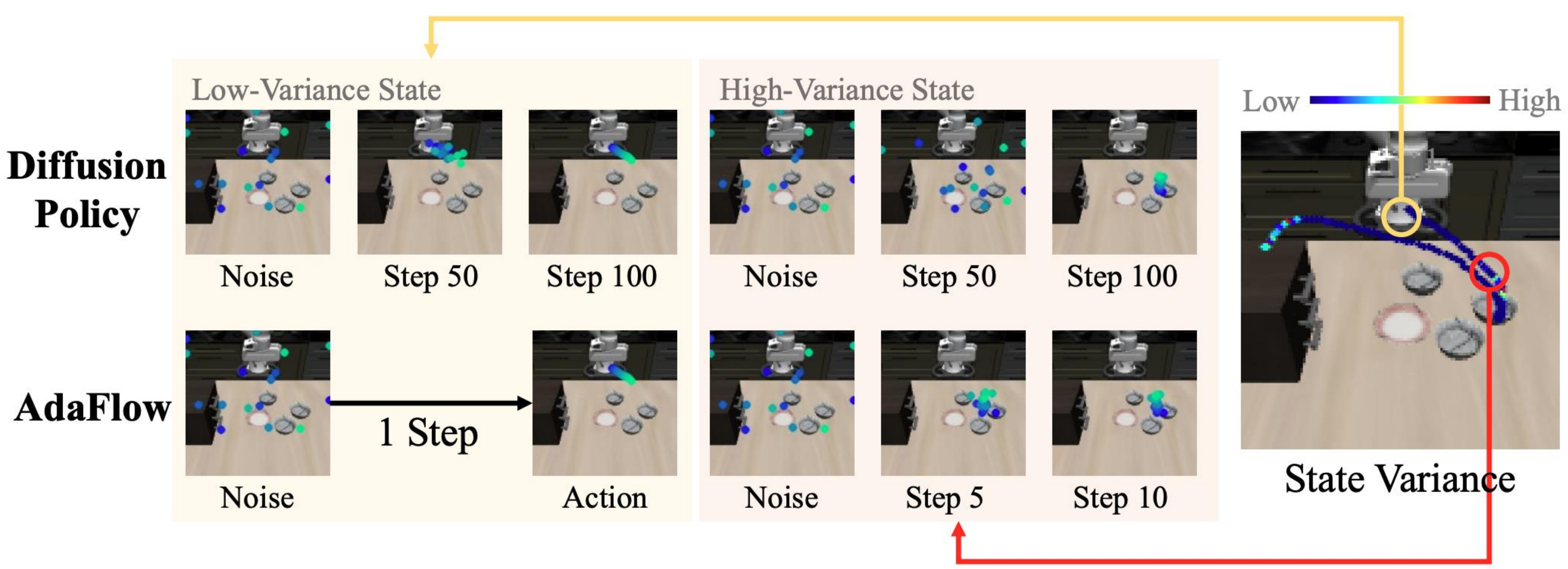
$$\epsilon_t = \mathrm{Clip}\left(\frac{\eta}{\sigma_\phi(\boldsymbol{z}_t, t \mid \boldsymbol{s})},\quad [\epsilon_{\min},\ 1-t]\right).$$

5: Update $t \leftarrow t + \epsilon_t$, $\boldsymbol{z}_t \leftarrow \boldsymbol{z}_t + \epsilon_t v_\theta(\boldsymbol{z}_t, t \mid \boldsymbol{s})$.
6: **end while**
7: Execute action $a = \boldsymbol{z}_1$.

(almost) **One-Step Generative Robot Policy**

No Distillation Needed!

	BC	Diffusion Policy	Rectified Flow	AdaFlow
Behavior Diversity	✗	✓	✓	✓
Fast Action Generation	✓	✗	✓	✓
No Distillation / Reflow	✓	✓	✗	✓



Experiments:

- AdaFlow outperforms baseline methods like BC and Diffusion Policy in both navigation and manipulation tasks.
- AdaFlow excels in benchmarks like RoboMimic and maze navigation, with high success rates and diverse actions.

Method	NFE↓	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Average
Rectified Flow (*Needs reflow*)	1	0.90	0.82	0.98	0.82	0.82	0.96	0.88
Diffusion Policy	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Diffusion Policy	2	0.00	0.58	0.36	0.66	0.36	0.32	0.38
Diffusion Policy	20	0.94	**0.84**	**0.98**	0.78	0.82	0.92	0.88
AdaFlow	**1.27**	**0.98**	0.80	**0.98**	**0.82**	**0.90**	**0.96**	**0.91**

Table 4: Success Rate on LIBERO Benchmark. The highest success rate for each task are highlighted in **bold**.

SCAN ME