A APPENDIX

A.1 VISUALIZING PRIMARY MODE DISTRIBUTION

To provide a more intuitive understanding of the Primary Mode Policy π_1, we present a qualitative analysis of its behavior during evaluation episodes. Figure 1 visualizes the policy's outputs at selected keyframes from four representative simulation tasks. For each keyframe, the policy's input point cloud p_t is shown alongside a heatmap representing the predicted probability distribution, over the discrete modes in the VQ codebook. These visualizations reveal that the policy learns a structured and context-aware mapping from inputs to high-level action primitives. As the episode progresses and the observation changes (*e.g.* from approaching an object to making contact) the distribution of predicted modes shifts accordingly, concentrating probability mass on a sparse set of task-relevant modes.

A.2 MORE SIMULATION RESULTS

To further verify the generality and stability of PF-DAG in robotic manipulation tasks, this appendix supplements the quantitative experimental results of PF-DAG and mainstream baselines on more tasks under the Adroit, DexArt, and MetaWorld benchmarks. These tasks cover both low-DOF gripper control and high-DOF dexterous hand manipulation, including additional fine-grained operations and complex task categories. All experimental settings are consistent with Section 6 of the main text, including the data collection pipeline, training hyperparameters, and evaluation metrics. The results in Table 1 further confirm that PF-DAG maintains consistent performance advantages over baselines across tasks of varying complexities.

A.3 MEANFLOW DERIVATION

This section provides a detailed derivation of the training objective for our mode-conditioned MeanFlow policy, as mentioned in Section 3.4. The formulation is based on the principles introduced by MeanFlow (Geng et al., 2025), which models the *average velocity* of a generative path rather than the *instantaneous velocity*.

Let the path between a noise sample $z_0 \sim \mathcal{N}(0, I)$ and the target action residual Δa be defined by an interpolation z_r for a time variable $r \in [0, 1]$. The instantaneous velocity at time r is denoted by $v(z_r, r) = \frac{dz_r}{dr}$.

The core concept is to define an **average velocity field** $\bar{v}(z_r, \tau, r; o, m)$ over an arbitrary time interval $[\tau, r]$, where o is the observation and m is the selected primary mode. This field is formally defined as the displacement between two points on the path, divided by the time interval:

$$\bar{v}(z_r, \tau, r; o, m) \triangleq \frac{1}{r - \tau} \int_{\tau}^{r} v(z_s, s; o, m) ds, \tag{1}$$

where s is the integration variable for time. To make this definition amenable to training, we first rewrite it by clearing the denominator:

$$(r - \tau)\bar{v}(z_r, \tau, r; o, m) = \int_{\tau}^{r} v(z_s, s; o, m) ds. \tag{2}$$

Next, we differentiate both sides with respect to the end time r, treating the start time τ as a constant. Applying the product rule to the left-hand side and the Fundamental Theorem of Calculus to the right-hand side yields:

$$\frac{d}{dr}\left[(r - \tau)\bar{v}(z_r, \tau, r)\right] = \frac{d}{dr} \int_{\tau}^{r} v(z_s, s) ds, \tag{3}$$

$$\bar{v}(z_r, \tau, r) + (r - \tau)\frac{d}{dr}\bar{v}(z_r, \tau, r) = v(z_r, r). \tag{4}$$

For clarity, we have omitted the conditioning on (o, m) in the last two steps. Rearranging the terms, we arrive at the **MeanFlow Identity**, which establishes a fundamental relationship between the average and instantaneous velocities:

$$\bar{v}(z_r, \tau, r) = v(z_r, r) - (r - \tau)\frac{d}{dr}\bar{v}(z_r, \tau, r). \tag{5}$$

Table 1: Quantitative comparison of PF-DAG against baselines on more tasks from Adroit, DexArt, and MetaWorld benchmarks.

	Adroit			**DexArt**				**Meta-World (Easy)**	
Alg \ Task	Hammer	Door	Pen	Laptop	Faucet	Toilet	Bucket	Button Press	Button Press Topdown
Diffusion Policy	45±5	37±2	13±2	69±4	23±8	58±2	46±1	99±1	98±1
3D Diffusion Policy	**100±0**	62±4	43±6	83±1	63±2	**82±4**	46±2	**100±0**	**100±0**
PF-DAG (Ours)	**100±0**	**65±3**	**65±3**	**90±2**	**72±5**	**82±2**	**65±3**	**100±0**	**100±0**

	Meta-World (Easy)						
Alg \ Task	Button Press Topdown Wall	Button Press Wall	Coffee Button	Dial Turn	Door Close	Door Lock	Door Open
Diffusion Policy	96±3	97±3	99±1	63±10	**100±0**	86±8	98±3
3D Diffusion Policy	99±2	99±1	**100±0**	**66±1**	**100±0**	**98±2**	99±1
PF-DAG (Ours)	**100±0**	**100±0**	**100±0**	55±10	**100±0**	94±6	**100±0**

	Meta-World (Easy)						
Alg \ Task	Door Unlock	Drawer Close	Drawer Open	Faucet Close	Faucet Open	Handle Press	Handle Pull
Diffusion Policy	98±3	**100±0**	93±3	**100±0**	**100±0**	81±4	27±22
3D Diffusion Policy	**100±0**	**100±0**	**100±0**	**100±0**	**100±0**	**100±0**	53±11
PF-DAG (Ours)	**100±0**	**100±0**	**100±0**	**100±0**	**100±0**	**100±0**	**55±5**

	Meta-World (Easy)						
Alg \ Task	Handle Press Side	Handle Pull Side	Lever Pull	Plate Slide	Plate Slide Back	Plate Slide Back Side	Plate Slide Side
Diffusion Policy	**100±0**	23±17	49±5	83±4	99±0	**100±0**	**100±0**
3D Diffusion Policy	**100±0**	**85±3**	79±8	**100±1**	99±0	**100±0**	**100±0**
PF-DAG (Ours)	**100±0**	77±4	**80±6**	**100±0**	**100±0**	**100±0**	**100±0**

	Meta-World (Easy)					**Meta-World (Medium)**	
Alg \ Task	Reach	Reach Wall	Window Close	Window Open	Peg Unplug Side	Basketball	Bin Picking
Diffusion Policy	18±2	59±7	**100±0**	**100±0**	74±3	85±6	15±4
3D Diffusion Policy	24±1	68±3	**100±0**	**100±0**	**75±5**	**98±2**	**34±30**
PF-DAG (Ours)	**29±5**	**71±3**	**100±0**	**100±0**	74±6	**98±2**	30±15

	Meta-World (Medium)						
Alg \ Task	Box Close	Coffee Pull	Coffee Push	Hammer	Peg Insert Side	Push Wall	Soccer
Diffusion Policy	30±5	34±7	67±4	15±6	34±7	20±3	14±4
3D Diffusion Policy	42±3	87±3	94±3	76±4	69±7	49±8	18±3
PF-DAG (Ours)	**70±5**	**89±5**	**95±3**	**100±0**	**71±1**	**69±2**	**34±3**

	Meta-World (Medium)		**Meta-World (Hard)**				
Alg \ Task	Sweep	Sweep Into	Assembly	Hand Insert	Pick Out of Hole	Pick Place	Push
Diffusion Policy	18±8	10±4	15±1	9±2	0±0	0±0	30±3
3D Diffusion Policy	**96±3**	15±5	**99±1**	14±4	14±9	12±4	51±3
PF-DAG (Ours)	92±5	**48±2**	98±2	**21±4**	**29±5**	**69±2**	**75±2**

	Meta-World (Hard)	**Meta-World (Very Hard)**					
Alg \ Task	Push Back	Shelf Place	Disassemble	Stick Pull	Stick Push	Pick Place Wall	**Average**
Diffusion Policy	0±0	11±3	43±7	11±2	63±3	5±1	55.4
3D Diffusion Policy	0±0	17±10	69±4	27±8	97±4	35±8	72.5
PF-DAG (Ours)	**6±5**	**52±8**	**77±7**	**59±6**	**100±0**	**82±6**	**79.6**

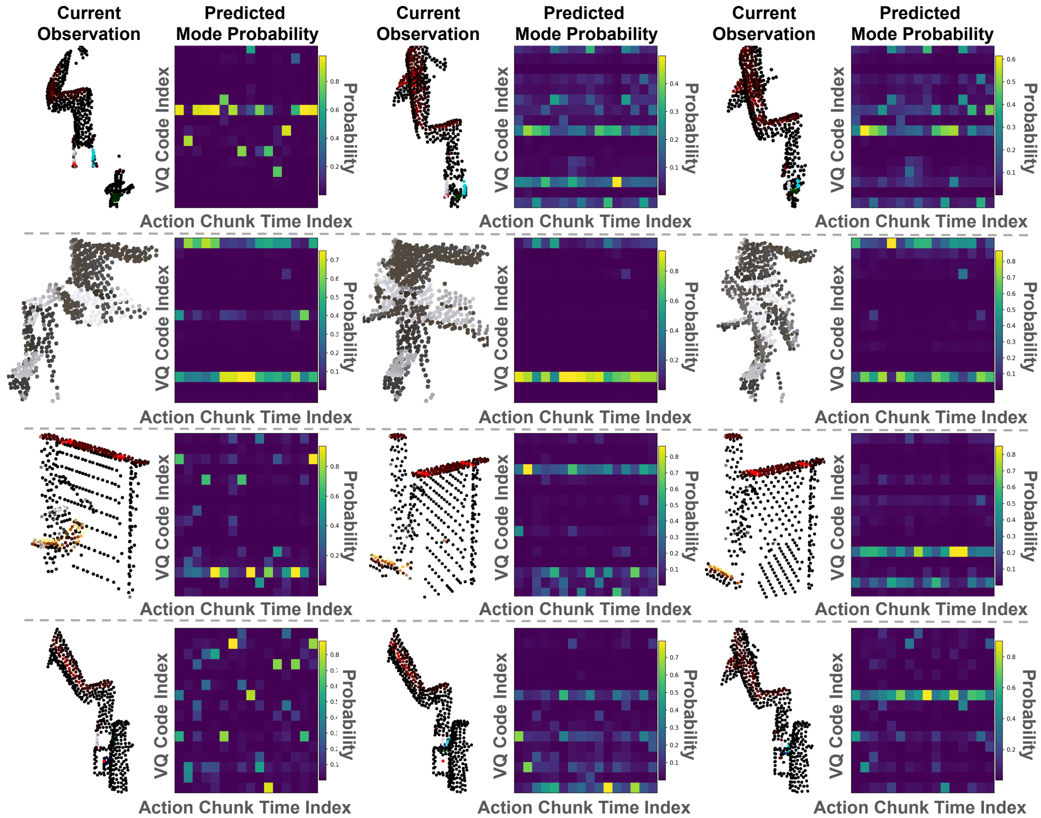

Figure 1: Qualitative visualization of the Primary Mode Policy (π_1) at keyframes from four different simulation tasks. Each row corresponds to a single task episode. Within each row, three keyframes show the point cloud observation (left) and the corresponding predicted probability distribution over the discrete primary modes (right) as a heatmap. The vertical axis of the heatmap represents the mode index. The shifting patterns in the heatmaps demonstrate that the policy learns a dynamic, context-dependent mapping from observation to a belief over high-level actions as the task progresses.

This identity provides a way to define a target for our neural network without computing an integral. To do so, we must first express the total time derivative $\frac{d}{dr}\bar{v}$ in a computable form. Since $\bar{v}$ is a function of (z_r, τ, r), we expand the total derivative using the chain rule:

$$\frac{d}{dr}\bar{v}(z_r, \tau, r) = \frac{\partial \bar{v}}{\partial z}\frac{dz_r}{dr} + \frac{\partial \bar{v}}{\partial \tau}\frac{d\tau}{dr} + \frac{\partial \bar{v}}{\partial r}\frac{dr}{dr}. \tag{6}$$

Given that $\frac{dz_r}{dr} = v(z_r, r)$, $\frac{d\tau}{dr} = 0$ (as τ is independent of r), and $\frac{dr}{dr} = 1$, the expression simplifies to:

$$\frac{d}{dr}\bar{v}(z_r, \tau, r) = v(z_r, r)\frac{\partial \bar{v}}{\partial z} + \frac{\partial \bar{v}}{\partial r}. \tag{7}$$

Substituting this result (7) back into the MeanFlow Identity (5), we obtain an expression for the average velocity that only depends on the instantaneous velocity v and the partial derivatives of $\bar{v}$:

$$\bar{v}(z_r, \tau, r) = v(z_r, r) - (r - \tau)\left(v(z_r, r)\frac{\partial \bar{v}}{\partial z} + \frac{\partial \bar{v}}{\partial r}\right). \tag{8}$$

This equation forms the basis for our training objective. We parameterize the average velocity field with a neural network $\bar{v}_\theta(z_r, \tau, r; o, m)$. The right-hand side of the equation becomes the regression target, where we replace the true partial derivatives of $\bar{v}$ with those of our network $\bar{v}_\theta$. Following standard practice, we apply a stop-gradient operator, $\mathrm{sg}(\cdot)$, to the target to prevent backpropagation through the Jacobian-vector products, which stabilizes training.

The resulting target, $\bar{v}_{tgt}$, is:

$$\bar{v}_{tgt} = v(z_r, r) - (r - \tau)\left(v(z_r, r)\frac{\partial \bar{v}_\theta}{\partial z} + \frac{\partial \bar{v}_\theta}{\partial r}\right). \tag{9}$$

The instantaneous velocity $v(z_r, r)$ is substituted with the conditional velocity (i.e., the ground-truth residual Δa minus the initial noise z_0). The final loss function is the expected squared ℓ_2 error between our network's prediction and this supervised target:

$$\mathcal{L}(\theta) = \mathbb{E}_{\Delta a, z_0, \tau, r} \left\| \bar{v}_\theta(z_r, \tau, r; o, m) - \text{sg}(\bar{v}_{tgt}) \right\|_2^2 . \tag{10}$$

This objective allows the network $\bar{v}_\theta$ to learn the average velocity field directly, enabling efficient one-step generation of the action residual Δa at inference time.

A.4 Implementation and Training Hyperparameters

This subsection details the key hyperparameters used for training and implementing our PF-DAG.

Hyperparameter	**Description**	**Value**
Prediction Horizon T_p	The total number of timesteps in a predicted action chunk.	32 / 16
Execution Horizon T_a	The number of timesteps from the chunk executed before re-planning.	16 / 8
Learning Rate	The peak learning rate after the warmup phase.	1e-4
Weight Decay	The weight decay value for the AdamW optimizer.	0.01
Batch Size	The number of samples processed per training step.	128
Codebook Size K	The number of discrete primary modes in the VQ-VAE codebook.	64
Commitment Weight β	The weight of the commitment loss term in the VQ-VAE objective.	0.25
VQ-VAE Latent Dim	The dimensionality of the VQ-VAE latent space.	64

Table 2: Hyperparameters for the PF-DAG framework.

A.5 Ablation Study on Mode Number

We present more results on mode number K, as seen in Table 3.

Ablations	Benchmarks			
# Modes	Adroit (3)	DexArt (4)	MetaWorld (11)	Weighted Success
64	$0.77_{\pm 0.03}$	$\mathbf{0.72}_{\pm \mathbf{0.04}}$	$\mathbf{0.70}_{\pm \mathbf{0.02}}$	**0.72**
8	$0.72_{\pm 0.03}$	$0.70_{\pm 0.02}$	$0.55_{\pm 0.05}$	0.61
16	$\mathbf{0.79}_{\pm \mathbf{0.02}}$	$0.71_{\pm 0.03}$	$0.68_{\pm 0.01}$	0.70
32	$0.76_{\pm 0.03}$	$0.71_{\pm 0.02}$	$0.67_{\pm 0.05}$	0.70
128	$0.77_{\pm 0.01}$	$0.69_{\pm 0.03}$	$0.67_{\pm 0.03}$	0.69
1024	$0.66_{\pm 0.02}$	$0.68_{\pm 0.03}$	$0.52_{\pm 0.06}$	0.58

Table 3: Ablation study on the mode number K of PF-DAG.

A.6 Quantitative Stability Analysis

To quantitatively validate our claim that PF-DAG produces more stable trajectories by reducing mode bouncing, we analyze the **total end-effector jerk** in our real-world experiments (Section 4.2), where stability is critical. Jerk, a standard metric for motion smoothness, is the integral of the squared magnitude of the third derivative of position over the trajectory duration T:

$$\text{Jerk} = \int_0^T \left\| \frac{d^3 \mathbf{p}(t)}{dt^3} \right\|^2 dt$$

A lower total jerk indicates a physically smoother, less shaky, and more stable trajectory. We computed this metric for the contact-rich 'Wipe Table' task from our real-world evaluation, comparing PF-DAG against DP3. As shown in Table A.6, PF-DAG achieves significantly lower jerk, confirming it generates smoother, less erratic end-effector movements.

Method	Total Jerk (↓)
DP3	1.25
PF-DAG (Ours)	**0.45**

Table 4: Total end-effector jerk (↓) comparison on the real-world 'Wipe Table' task.

A.7 Rigorous Analysis of the MSE Trade-off

The analysis in the main text assumes an oracle π_1 to illustrate how our architecture decomposes variance. Here, we provide a more rigorous analysis of the practical trade-off, considering errors from our learned π_1.

A.7.1 The Problem with MSE-Optimal Predictors

The central thesis of our paper is that in multi-modal tasks, a predictor that is "optimal" under the Mean Squared Error (MSE) criterion is undesirable. A standard Behavioral Cloning (BC) model that predicts the conditional expectation $\hat{a}^*(o) = \mathbb{E}[a|o]$ is, by definition, the optimal deterministic predictor. Its minimum achievable loss is:

$$L_g^* = \mathbb{E}_o[Var(a|o)]$$

Using the law of total variance, we decompose this loss:

$$L_g^* = \underbrace{\mathbb{E}_{o,m}[Var(a|o,m)]}_{V_{\text{intra}}} + \underbrace{\mathbb{E}_o[Var_{m|o}(\mathbb{E}[a|o,m])]}_{V_{\text{inter}}}$$

- V_{intra}: The **within-mode variance**. This is the fine-grained variation that our π_2 must model.
- V_{inter}: The **inter-mode variance**. This is the variance between the means of the different modes (*e.g.*, the difference between "go left" and "go right").

The MSE-optimal predictor $\mathbb{E}[a|o]$ averages these modes, resulting in V_{inter} as a fundamental component of its error. This is precisely **mode collapse**, which is catastrophic for task success.

A.7.2 The PF-DAG Trade-off: V_{INTER} vs. E_{CLASSIFY}

Our two-stage model, PF-DAG, makes a "hard" mode selection $\hat{m} = \pi_1(o)$. The final action is the prediction of the second stage, $\hat{a}_{\text{PF-DAG}}(o) = \mathbb{E}_z[\pi_2(o, \hat{m}, z)]$. For this analysis, let's assume a perfect π_2 that correctly predicts the mean of its target mode, *i.e.*, $\mathbb{E}_z[\pi_2(o,k,z)] = \mathbb{E}[a|o, m=k]$, which we denote $\mu_k(o)$. The practical MSE of our model is $L_{\text{PF-DAG}} = \mathbb{E}_{o,a}[||a - \mu_{\hat{m}(o)}(o)||^2]$. We decompose this by conditioning on the true, unobserved mode m:

$$L_{\text{PF-DAG}} = \mathbb{E}_o\left[\sum_m p(m|o)\mathbb{E}_{a|o,m}[||a - \mu_{\hat{m}(o)}(o)||^2]\right]$$

Using the identity $\mathbb{E}[||X-c||^2] = Var(X) + ||\mathbb{E}[X] - c||^2$, where $X = a|o,m$ and $c = \mu_{\hat{m}(o)}(o)$:

$$L_{\text{PF-DAG}} = \mathbb{E}_o\left[\sum_m p(m|o)(Var(a|o,m) + ||\mu_m(o) - \mu_{\hat{m}(o)}(o)||^2)\right]$$

$$L_{\text{PF-DAG}} = \underbrace{\mathbb{E}_{o,m}[Var(a|o,m)]}_{V_{\text{intra}}} + \underbrace{\mathbb{E}_o\left[\sum_m p(m|o)||\mu_m(o) - \mu_{\hat{m}(o)}(o)||^2\right]}_{E_{\text{classify}}}$$

This reveals the explicit trade-off of our architecture:

- **Single-Stage (BC):** $L_g^* = V_{\text{intra}} + V_{\text{inter}}$

- **PF-DAG (Ours):** $L_{\text{PF-DAG}} = V_{\text{intra}} + E_{\text{classify}}$

PF-DAG is designed to trade V_{inter} (the guaranteed, catastrophic cost of mode collapse) for E_{classify} (the probabilistic cost of misclassification). Our strong empirical task success (Tables 1, 2, 5) supports our hypothesis that V_{inter} is fatal for task execution, while E_{classify} is a non-catastrophic and manageable error. Our framework replaces a guaranteed failure mode with a high-probability success, which is a highly desirable trade-off for robotic imitation.

A.8 Acknowledgments on LLM Usage

We acknowledge the use of a large language model (LLM) for aiding in the writing and polishing of this paper. The LLM is used as a tool to improve the clarity, grammar, and style of certain sections. Its contributions are limited to editorial and linguistic improvements, and it is not used to generate novel ideas, perform research, or formulate the core technical content. All scientific contributions, experimental results, and intellectual content are the original work of the human authors.

References

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.