BILINEAR FLOW POLICY: DISTRIBUTIONAL EXTRAPOLATION FOR GOAL-CONDITIONED VISUOMOTOR IMITATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Goal-conditioned imitation learning (GCIL) with flow matching is a promising framework that can represent multimodal behaviors while adapting to diverse, user-specified goals, yet often fails when goals lie outside the demonstration support. To extrapolate to such unseen goals without collapsing multimodality—a problem we call *distributional extrapolation*—we introduce *Bilinear Flow Policy* (BFP), a generative visuomotor policy that combines transductive retrieval with a bilinear conditional flow. Given an unseen observation-goal pair, BFP retrieves an "anchor" training example and transductively reformulates the unseen pair as this familiar anchor plus a residual term. For this decomposition to guide action prediction, the residual must compactly encode how the current observation-goal pair differs from the anchor, and the anchor must be chosen so that this difference is predictive of the corresponding action distribution. BFP achieves this with pretrained visual features and a novel learned anchor-selection algorithm. The novel bilinear flow then models how the anchor and the residual jointly determine the multimodal action distribution. We prove that, for bilinear flow, action distribution error at unseen goals is bounded by the in-distribution flow-matching error up to problem-dependent factors. Across five manipulation tasks in simulation, BFP achieves $2.63\times$ the out-of-distribution success rate of a GCIL policy and $1.36\times$ that of the strongest extrapolation-targeted baseline. On two real-world tasks, BFP improves over GCIL by 32 percentage points on average. Finally, our theory yields practical, pre-deployment diagnostics for predicting which trained policies will extrapolate well and to which unseen goals. Project website: `https://bilinearflow.netlify.app`

1 INTRODUCTION

Goal-conditioned imitation learning (GCIL) is a promising framework to learn versatile visuomotor behavior: a single policy can adapt to diverse, user-specified goals at deployment (Ding et al., 2019; Park et al., 2025). Conditional flow matching further makes this framework expressive by modeling multimodal distributions of actions, allowing the policy to represent multiple behavior modes to achieve the same goal (Reuss et al., 2023; 2024). Nevertheless, deployment may require the policy to act toward unseen goals, where it encounters observation–goal pairs not seen during training. Standard GCIL policies, including generative variants, can become unreliable on such unseen inputs (Yang et al., 2023; Urain et al., 2024). This motivates *distributional extrapolation*: extending goal-conditioned behavior to unseen goals while preserving multimodality.

One natural route to extrapolation is to identify how behavior should change as the policy input (i.e., current observation and goal) changes, then reuse the same rule to infer behavior for a new input by comparing it with a training example (Ousherovitch & Wang, 2026). When this relation is known, it can be encoded directly into the policy; for instance, the behavior should remain invariant to task-irrelevant perceptual perturbations (Zhu et al., 2023b), transform equivariantly under known geometric changes (Wang et al., 2024a), or be composed of reusable policy components (Mishra et al., 2023). These methods generalize by encoding a particular, known deployment shift as an inductive bias. They are effective when this prescribed relation matches the nature of the shift, but require the relevant task-specific semantics or transformation laws to be specified a priori.

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.

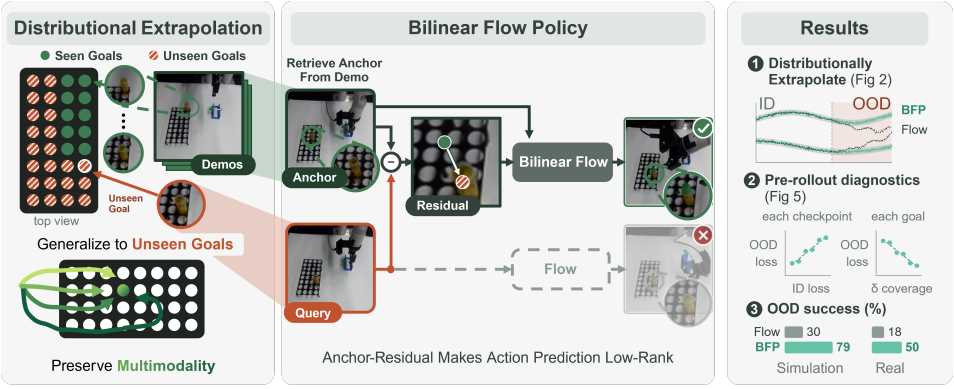


Figure 1: **BFP Overview.** BFP learns how changes across observation-goal conditions affect behavior and reuses this relation to extrapolate from a demonstrated anchor to an unseen goal. A bilinear conditional flow generates the resulting multimodal action distribution, while the explicit extrapolation rule enables theoretical guarantees and practical diagnostics.

When the underlying observation-goal-action relation is unknown, an attractive alternative is to *learn* it from comparisons among demonstrations. Bilinear transduction (Netanyahu et al., 2023) achieves this by decomposing an observation-goal condition x into an "anchor" training example a and a residual $\delta = x - a$, then constrains the predictor to a low-rank interaction between features of the two factors. Thus, rather than specifying a task-specific transformation explicitly, the model learns from the data how the anchor and the residual jointly determine behaviors within a generic low-rank relationship. This reparameterization converts an out-of-support query into a potentially unseen combination of individually familiar factors; under low-rank, coverage, and conditioning assumptions, prediction error at unsupported queries is provably bounded by the training error (Shah et al., 2020; Simchowitz et al., 2023). Existing robotic instantiations, however, are state-based and deterministic, leaving open how to construct this decomposition from visuomotor demonstrations and how to extrapolate a multimodal action distribution.

In this work, we ask whether this relational formulation can support generative goal-conditioned visuomotor policies. We introduce *Bilinear Flow Policy* (BFP), which extrapolates to unseen goals by transductively relating novel observation-goal conditions to demonstrated examples in the training data. BFP applies bilinear transduction to a conditional flow field, extending relational extrapolation from deterministic actions to multimodal action distributions. It also combines a compact visual representation with a learned anchor policy to find generalizable relations in visuomotor settings.

We also derive an end-to-end distributional extrapolation bound that relates in-distribution flow-matching error to the Wasserstein error of the generated action distribution at unseen goals. The bound's training-error and coverage terms further motivate two pre-deployment diagnostics: held-out flow loss for selecting among trained policies, and anchor–residual coverage for identifying unseen-goal queries on which a selected policy is likely to remain reliable.

We evaluate BFP on five goal-conditioned visuomotor manipulation tasks across three simulation benchmarks. On the simulation tasks, BFP achieves 79.6% average out-of-distribution (OOD) success ($2.63\times$ that of matched generative GCIL and $1.36\times$ that of the strongest extrapolation-targeted baseline) while retaining comparable in-distribution performance, across 4,200 rollouts. On the two real-world robotics tasks, BFP improves over the matched GCIL baseline by 32 percentage points on average.

In summary, our contributions are threefold.

- We introduce BFP, a generative goal-conditioned visuomotor imitation policy learning framework that extrapolates to unseen goals preserving behavior multimodality.
- We derive an end-to-end bound relating in-distribution flow error to unseen-goal action-distribution error. The theorem motivates practical pre-deployment diagnostics for selecting trained policies and assessing their reliability on specific unseen goals.

- Across controlled multimodal regression, five simulated manipulation tasks, and two real-world tasks, BFP substantially improves OOD performance over matched generative GCIL and extrapolation-targeted baselines.

2 Related Work

We first situate our work within the generalization regimes studied in goal-conditioned policies. We then categorize extrapolation methods by how they map changes in policy inputs into behavior changes and reuse that rule beyond training support. Prior work either hand-crafts this rule or use demonstration data to learn it. BFP follows the latter approach.

Generalization Regimes of Goal-Conditioned Policies. Goal-conditioned policies use a single policy across many desired outcomes (Gupta et al., 2019; Park et al., 2025), while generative variants model multimodal action distributions for a given goal (Reuss et al., 2023; 2024). Their generalization beyond training state–goal combinations has been studied under two distinct regimes. In *stitching*, where both the state and goal have been seen during training, but not together, the policy must recombine familiar transitions or trajectory segments to connect them (Lawson et al., 2026; Luo et al., 2026). Another studies *unseen-goal generalization*, where the desired goal itself lies outside the training support and pursuing it requires the policy to act at out-of-distribution state–goal inputs (Hong et al., 2022; Yang et al., 2023). We study this regime in generative visuomotor imitation, where policies must generalize from visual inputs while preserving multimodal behavior.

Hand-crafted Extrapolation Rules. Some approaches extrapolate through hand-specified rules for how behavior changes with the policy input (Urain et al., 2024). Representation-based methods seek invariance to task-irrelevant variation by reducing observations to task-relevant objects, regions, keypoints, or geometry (Zhu et al., 2023b; Manuelli et al., 2019; Zhu et al., 2023a). Equivariant policies prescribe how known scene transformations should transform actions or action distributions (Wang et al., 2024a; Funk et al., 2024). Compositional methods define reusable factors (e.g., skills, objectives, or sensor modalities) and prescribe how they may be recombined (Mishra et al., 2023; Urain et al., 2023; Wang et al., 2024b). Their representations and components may be learned, but the rule is fixed a priori; this is effective when relevant structure is known, but restrictive otherwise. We instead study settings where this rule is not obvious and must be inferred from demonstrations.

Learning Extrapolation Rules from Demonstrations. When no rule is known *a priori*, some prior works learn it by predicting a desired action from training examples and residuals that quantify the input change from each example to the query. DARP (Pfeifer et al., 2026) instantiates this locally by retrieving the k nearest examples, predicting from each example and its residual, and aggregating the results. Bilinear transduction (BT) (Netanyahu et al., 2023) instead assumes that a suitable anchor and compact residual induce a low-rank action-prediction problem, and uses a bilinear model to predict actions for unseen anchor–residual combinations. Subsequent work extends this principle to offline RL and task-level extrapolation (Kim et al., 2026; Ousherovitch & Wang, 2026). Robotic applications, however, have considered state inputs, where residuals are directly defined in compact coordinates and anchors are chosen using task-specific heuristics. We study whether the same principle extends to generative visuomotor imitation, where defining a compact residual, selecting an anchor that exposes the low-rank relation, and preserving multiple valid actions are all nontrivial.

3 Problem Formulation and Preliminaries

Problem Formulation: Distributional Extrapolation. Let $\mathbb{A} \subset \mathbb{R}^{d_a}$ be the primitive action space and $\mathbb{A}_H := \mathbb{A}^H \subset \mathbb{R}^{d_A}$, $d_A = Hd_a$ be the action-chunk space. We are given expert demonstrations: $\mathcal{D} = \{\tau^{(i)}\}_{i=1}^{N}, \tau^{(i)} = \{(\boldsymbol{x}_t^{(i)}, \boldsymbol{a}_t^{(i)})\}_{t=1}^{T_i}$, where $\boldsymbol{a}_t^{(i)} \in \mathbb{A}$ is a primitive action, and $\boldsymbol{x}_t^{(i)} = (\boldsymbol{o}_t^{(i)}, \boldsymbol{g}^{(i)}) \in \mathbb{X}$ is the policy condition. Here $\boldsymbol{o}_t^{(i)}$ is a state or visuomotor observation, and $\boldsymbol{g}^{(i)}$ is a compact, explicitly specified episode goal (e.g., target object pose; See Fig. 1). Each timestep defines an H-step action-chunk target $\boldsymbol{A}_t^{(i)} := (\boldsymbol{a}_t^{(i)}, \dots, \boldsymbol{a}_{t+H-1}^{(i)}) \in \mathbb{A}_H$.

Let P_{tr}^X and P_{te}^X denote the marginal distribution of policy conditions under training and evaluation, respectively; we refer to conditions within the demonstrated support as in-distribution (ID) and those outside it as out-of-distribution (OOD). We assume the same expert conditional $q^\star(\cdot \mid \boldsymbol{x})$ in both

regimes, while P_{te}^X may include conditions induced by unseen goals outside $\text{supp}(P_{\text{tr}}^X)$. Given only $\mathcal{D}$, we seek a policy $\boldsymbol{\pi}_\theta$ with small test distributional risk

$$\mathcal{R}_{\text{te}}^{\text{dist}}(\boldsymbol{\pi}_\theta) := \mathbb{E}_{\boldsymbol{x}\sim P_{\text{te}}^X}\left[W_2^2(\boldsymbol{\pi}_\theta(\cdot \mid \boldsymbol{x}), q^\star(\cdot \mid \boldsymbol{x}))\right]. \tag{1}$$

where W_2^2 is the squared 2-Wasserstein distance (Peyré & Cuturi, 2019).

We call this problem *distributional extrapolation*; a policy learned from demonstrated conditions must match the full expert action distribution at unseen conditions. This matters when a condition admits multiple action chunks: as illustrated in Fig. 2, mean regression is vulnerable to mode-averaging, while distributional modeling alone does not specify how the expert modes should extend beyond training support.

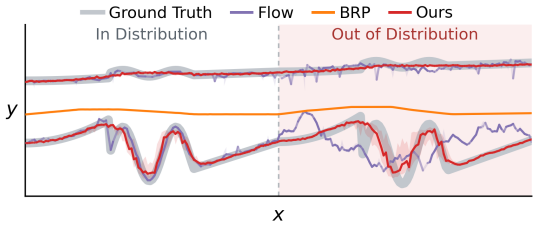


Figure 2: **Distributional extrapolation.** The expert conditional distribution $q^\star(y \mid x)$ is a mixture of two modes (grey), evaluated outside $\text{supp}(P_{\text{tr}}^X)$ (shaded red). BT averages the modes (orange), while CFM captures multimodality but not extrapolation (purple). BFP achieves both (red).

Conditional Flow Matching. Conditional flow matching (CFM) (Lipman et al., 2024) transports a simple prior $\boldsymbol{A}_0 \sim p_0$ (e.g., Gaussian) to the expert action chunk $\boldsymbol{A}_1 = \boldsymbol{A} \sim q^\star(\cdot \mid \boldsymbol{x})$. For a conditional probability path $p_\tau(\boldsymbol{A}_\tau \mid \boldsymbol{A})$ with corresponding velocity field $u_\tau(\boldsymbol{A}_\tau \mid \boldsymbol{A})$, CFM minimizes

$$\mathcal{L}_{\text{CFM}}(\theta) := \mathbb{E}_{(\boldsymbol{x},\boldsymbol{A})\sim\mathcal{D},\tau\sim\mathcal{U}[0,1],\boldsymbol{A}_\tau\sim p_\tau(\cdot\mid\boldsymbol{A})}\left[\left\|v_\theta(\boldsymbol{A}_\tau,\tau \mid \boldsymbol{x}) - u_\tau(\boldsymbol{A}_\tau \mid \boldsymbol{A})\right\|_2^2\right]. \tag{2}$$

Under squared loss, the regression target is

$$u_\tau^\star(\boldsymbol{y} \mid \boldsymbol{x}) := \mathbb{E}[u_\tau(\boldsymbol{A}_\tau \mid \boldsymbol{A}) \mid \boldsymbol{A}_\tau = \boldsymbol{y}, \boldsymbol{x}]. \tag{3}$$

At test-time, integrating $\dot{\boldsymbol{A}}_\tau = v_\theta(\boldsymbol{A}_\tau, \tau \mid \boldsymbol{x})$ from $\boldsymbol{A}_0 \sim p_0$ generates an action chunk.

Bilinear Transduction (BT). To extrapolate, BT (Netanyahu et al., 2023) selects a training example $\boldsymbol{x}^\dagger \in \mathcal{D}$ as an anchor for a query condition $\boldsymbol{x}$, defines the residual $\boldsymbol{\delta} := \boldsymbol{x} - \boldsymbol{x}^\dagger$, and predicts the target through a low-rank bilinear interaction between the anchor and residual. Specifically, it reparameterizes a regression target $h^\star(\boldsymbol{x}) \in \mathbb{R}^{d_A}$ as $\widetilde{h}^\star(\boldsymbol{\delta}, \boldsymbol{x}^\dagger) := h^\star(\boldsymbol{x}^\dagger + \boldsymbol{\delta})$ and models

$$\widetilde{h}_\theta(\boldsymbol{\delta}, \boldsymbol{x}^\dagger) = \left[\left\langle f_{\theta,k}(\boldsymbol{\delta}), g_{\theta,k}(\boldsymbol{x}^\dagger)\right\rangle_r\right]_{k=1}^{d_A} \qquad f_{\theta,k}(\boldsymbol{\delta}),\ g_{\theta,k}(\boldsymbol{x}^\dagger) \in \mathbb{R}^r, \tag{4}$$

where $\langle\cdot,\cdot\rangle_r$ denotes an inner product over a low-dimensional rank-r latent space. Together with the anchor-selection rule, this predictor induces training and test risks $\mathcal{R}_{\text{tr}}$ and $\mathcal{R}_{\text{te}}$, measured by the squared prediction error $\|\widetilde{h}_\theta - \widetilde{h}^\star\|_2^2$. BT then admits the following extrapolation bound:

$$\mathcal{R}_{\text{te}}(\widetilde{h}_\theta) \le C_{\text{BT}}\,\mathcal{R}_{\text{tr}}(\widetilde{h}_\theta), \tag{5}$$

where C_{BT} is a problem-dependent factor, under two requirements: **(S) Bilinear Structure** the anchor must be selected, and the residual defined, so that the resulting anchor–residual-to-action prediction problem is approximately low-rank. **(C) Residual Coverage** the test-time anchor should yield a residual covered during training.

Limitations. BT addresses **(C) Residual Coverage** through test-time anchor filtering, but largely takes **(S) Bilinear Structure** for granted. Consequently, prior demonstrations are limited to specific tasks in low-dimensional state spaces, where residuals are meaningful and anchor selection rules can be hand-designed. Real-world visuomotor imitation introduces two hurdles. First, inducing **(S) Bilinear Structure** requires finding both a representation with meaningful residuals and a retrieval policy that yields a simple prediction problem. Second, expert actions may be multimodal, whereas BT predicts deterministically. BFP addresses both challenges in the next section.

4 Method

Overview. Fig. 3 illustrates our proposed *Bilinear Flow Policy* (BFP), a generative goal-conditioned visuomotor policy that extrapolates to unseen goals while preserving behavior multimodality. BFP

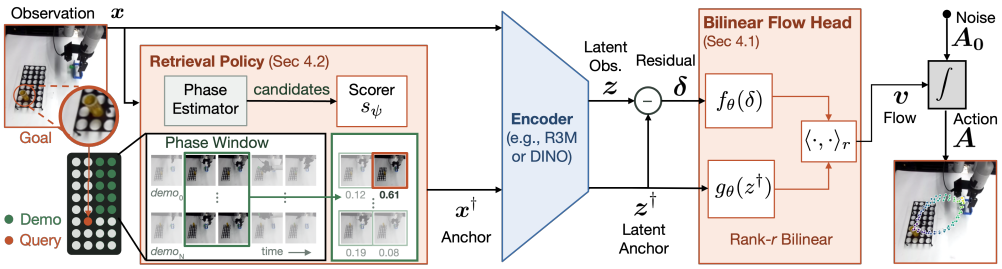


Figure 3: **Overview of BFP.** Given an unseen observation–goal condition $\boldsymbol{x}$, BFP retrieves an anchor $\boldsymbol{x}^\dagger$ and predicts from the resulting anchor–residual pair. The novel retrieval policy (Sec. 4.2) uses a phase estimator to form a phase-compatible candidate set and a learned scorer s_ψ, trained from held-out BFP loss, to select $\boldsymbol{x}^\dagger$. A frozen visual encoder computes $\boldsymbol{z} = \phi(\boldsymbol{x})$ and $\boldsymbol{z}^\dagger = \phi(\boldsymbol{x}^\dagger)$, with residual $\boldsymbol{\delta} = \boldsymbol{z} - \boldsymbol{z}^\dagger$. The novel bilinear conditional-flow predictor (Sec. 4.1) predicts the CFM velocity through the low rank-r interaction; flow integration then generates the action chunk.

addresses the two challenges through a pretrained representation, learned anchor retrieval, and a bilinear conditional flow model. Formally, given a query $\boldsymbol{x} = (\boldsymbol{o}, \boldsymbol{g})$, BFP retrieves an anchor $\boldsymbol{x}^\dagger$ from the demonstrations, encodes both conditions with a latent representation ϕ, then forms

$$\boldsymbol{z} = \phi(\boldsymbol{x}), \qquad \boldsymbol{z}^\dagger = \phi(\boldsymbol{x}^\dagger), \qquad \boldsymbol{\delta} = \boldsymbol{z} - \boldsymbol{z}^\dagger. \tag{6}$$

The goal is to choose a representation and anchor policy so that the retrieved anchor and its residual makes the required action at query *straightforward to predict*; concretely, we seek an induced anchor–residual–action prediction problem with low-rank bilinear structure (i.e., **(S) Bilinear Structure**). Sec. 4.1 first assumes this structure and shows that imposing it to the CFM marginal velocity field yields distributional extrapolation. Sec. 4.2 then constructs ϕ and ρ_ψ to make this structure plausible in visuomotor demonstrations.

4.1 Bilinear Conditional Flow Head

We first isolate the predictor question. Given a representation ϕ and retrieval policy ρ_ψ that induce a low-rank relation between the retrieved anchor, residual, and desired action, how can we achieve *distributional extrapolation*?

The representation ϕ and anchor policy ρ_ψ re-express the expert conditional $q^\star(\cdot \mid \boldsymbol{x})$ in the anchor–residual coordinates of Eq. (6) as $q^\star(\cdot \mid \boldsymbol{\delta}, \boldsymbol{z}^\dagger)$. Together with the conditional probability path $p_\tau(\boldsymbol{A}_\tau \mid \boldsymbol{A})$, this defines the marginal velocity field $u^\star_\tau(\boldsymbol{A}_\tau \mid \boldsymbol{\delta}, \boldsymbol{z}^\dagger)$ in Eq. (3), whose flow generates the same expert conditional.

BFP approximates this field with the coordinate-wise bilinear model

$$v_{\theta,k}(A_\tau, \tau \mid \boldsymbol{\delta}, \boldsymbol{z}^\dagger) = \left\langle f_{\theta,k}(\boldsymbol{\delta}; A_\tau, \tau), g_{\theta,k}(\boldsymbol{z}^\dagger; A_\tau, \tau) \right\rangle_r, \qquad k \in [d_A]. \tag{7}$$

For each fixed flow context $c = (A_\tau, \tau)$ and action coordinate k, this assumes that the marginal velocity field can be approximated by a rank-r bilinear function of the residual and anchor. Thus, **(S) Bilinear Structure** is imposed on the velocity field rather than directly on the action.

We train this predictor with the CFM objective. For a demonstration $(\boldsymbol{x}, A)$ and retrieved anchor $\boldsymbol{x}^\dagger$,

$$\ell_{\mathrm{BFP}}(\theta; \boldsymbol{x}, \boldsymbol{x}^\dagger, A) := \mathbb{E}_{\substack{\tau \sim \mathcal{U}[0,1] \\ A_\tau \sim p_\tau(\cdot \mid A)}} \left[\left\| v_\theta(A_\tau, \tau \mid \boldsymbol{\delta}, \boldsymbol{z}^\dagger) - u_\tau(A_\tau \mid A) \right\|_2^2 \right]. \tag{8}$$

We also define $\mathcal{E}_s^{\mathrm{flow}}(\theta) := \mathbb{E}_s \left[\| v_\theta - u^\star \|_2^2 \right]$ for $s \in [\mathrm{tr}, \mathrm{te}]$. We now present our main result:

Theorem 1 (Distributional extrapolation of BFP, informal)**.** *Suppose that the marginal velocity field satisfies **(S) Bilinear Structure**, and test-time anchor selection satisfies **(C) Residual Coverage**. Under the remaining technical conditions detailed in Appx. C.3.2, sufficiently small training flow error implies*

$$\mathcal{R}^{\mathrm{dist}}_{\mathrm{te}}(\pi_\theta) \le C_{\mathrm{ODE}}\, \mathcal{E}^{\mathrm{flow}}_{\mathrm{te}}(\theta) \le C_{\mathrm{ODE}} C_{\mathrm{BFP}}\, \mathcal{E}^{\mathrm{flow}}_{\mathrm{tr}}(\theta). \tag{9}$$

Proof. Deferred to Appx. C.3.2. □

Practical interpretation. Thus, in-distribution flow error controls the full generated action distribution at unseen goals, provided that the representation and retrieval policies expose a low-rank and covered factorization. Furthermore, the theorem motivates the assumption of this theorem quantifying **(S) Bilinear Structure** and **(C) Residual Coverage** as the diagnostics of extrapolation performance. Their diagnostics are evaluated in Sec. 5.3.

4.2 Learning BFP from Demonstrations

We now instantiate the representation and anchor policies left abstract in Sec. 4.1: training-time retrieval aims to induce a simple low-rank relation (**(S) Bilinear Structure**), while test-time retrieval aims to keep the required residual covered by training (**(C) Residual Coverage**).

Representation. Let $\boldsymbol{x} = (I^{1:V}, \boldsymbol{p}, \boldsymbol{g})$ contain images, proprioception, and the goal. BFP uses

$$\phi(\boldsymbol{x}) = \left[E_{\text{vis}}(I^1); \ldots; E_{\text{vis}}(I^V); \boldsymbol{p}; \boldsymbol{g}\right], \tag{10}$$

where E_{vis} is a frozen pretrained visual representation such as R3M (Nair et al., 2022) or DINOv3 (Siméoni et al., 2025). Applying the same encoder to query and anchor provides a shared compact space for defining $\boldsymbol{\delta} = \phi(\boldsymbol{x}) - \phi(\boldsymbol{x}^\dagger)$. As it is pretrained independent to demonstrations, they provide a common train–test coordinate system.

Learning train-time anchor retrieval. Anchors at similar task phases are more likely to induce simple residuals, but phase compatibility alone does not guarantee bilinear predictability. For each query $\boldsymbol{x}$, a coarse phase estimator therefore forms a candidate window $\mathcal{C}(\boldsymbol{x}) = \{\boldsymbol{x}_m^\dagger\}_{m=1}^K$ from demonstrations at similar task progress. We use a simple hand-designed estimator, although learned temporal-alignment methods could be used instead (Dwibedi et al., 2019; Memmel et al., 2025).

For each candidate, let $\boldsymbol{z}_m^\dagger = \phi(\boldsymbol{x}_m^\dagger)$ and $\boldsymbol{\delta}_m = \phi(\boldsymbol{x}) - \boldsymbol{z}_m^\dagger$. A learned scorer defines

$$p_\psi(m \mid \boldsymbol{x}, \mathcal{C}(\boldsymbol{x})) = \frac{\exp\big(s_\psi(\boldsymbol{\delta}_m, \boldsymbol{z}_m^\dagger)/\tau_\rho\big)}{\sum_{n=1}^K \exp\Big(s_\psi(\boldsymbol{\delta}_n, \boldsymbol{z}_n^\dagger)/\tau_\rho\Big)}. \tag{11}$$

We alternate predictor and scorer updates so that each provides the learning signal for the other. With the scorer fixed, the predictor learns primarily from anchors assigned high probability:

$$\mathcal{L}_{\text{pred}}(\theta) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{A})\sim\mathcal{D}}\left[\sum_{m=1}^K \text{sg}[p_\psi(m \mid \boldsymbol{x}, \mathcal{C}(\boldsymbol{x}))]\ell_{\text{BFP}}(\theta; \boldsymbol{x}, \boldsymbol{x}_m^\dagger, \boldsymbol{A})\right]. \tag{12}$$

Conversely, with the predictor fixed, the scorer favors anchors with low held-out BFP loss:

$$\mathcal{L}_{\text{ret}}(\psi) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{A})\sim\mathcal{D}_{\text{val}}}\left[\sum_{m=1}^K p_\psi(m \mid \boldsymbol{x}, \mathcal{C}(\boldsymbol{x}))\text{sg}[\widetilde{\ell}_m]\right], \tag{13}$$

where $\widetilde{\ell}_m$ is the normalized held-out BFP loss. Thus, retrieval favors anchor–residual pairs that the shared rank-r predictor can explain well. Algorithm and details are given in Appx. D.

Test-time anchor selection. At test time, we instead select anchors to satisfy **(C) Residual Coverage**. Following BT, we first select demonstrations whose goal residual is closest to the empirical training goal-delta set $\widehat{\mathcal{G}}_\Delta$,

$$j^\star \in \arg\min_j \min_{\Delta g \in \widehat{\mathcal{G}}_\Delta} \left\|(g_q - g^{(j)}) - \Delta g\right\|_2^2. \tag{14}$$

We apply the learned scorer from Eq. (11) over $\mathcal{C}_{\text{te}}(\boldsymbol{x})$ and sample $M \sim \text{Cat}(p_\psi(\cdot \mid \boldsymbol{x}, \mathcal{C}_{\text{te}}(\boldsymbol{x})))$. With the sampled anchor fixed, we form $\boldsymbol{z}_M^\dagger = \phi(\boldsymbol{x}_M^\dagger)$ and $\boldsymbol{\delta}_M = \phi(\boldsymbol{x}) - \boldsymbol{z}_M^\dagger$, sample $\boldsymbol{A}_0 \sim p_0$, and integrate

$$\dot{\boldsymbol{A}}_\tau = v_\theta(\boldsymbol{A}_\tau, \tau \mid \boldsymbol{\delta}_M, \boldsymbol{z}_M^\dagger), \qquad \tau \in [0, 1]. \tag{15}$$

The terminal state $\boldsymbol{A}_1$ is the generated action chunk. Algorithm and details are given in Appx. D.

5 EXPERIMENTS

We organize our experiments around three questions that follow the claims of the method. First (Sec. 5.1), can BFP solve distributional extrapolation given oracle representation and retrieval? Second (Sec. 5.2), does BFP generalize to unseen goals and transfer to real-world deployment? Third (Sec. 5.3), can theory-motivated signals identify which trained policies and unseen queries are likely to extrapolate reliably without performing OOD rollouts? We additionally ablate core components of BFP (Sec. 5.4). Full results are reported in Appxs. A and B.

5.1 CAN BFP SOLVE DISTRIBUTIONAL EXTRAPOLATION?

Setting. We first isolate the predictor from the representation and anchor-retrieval challenges studied later. We consider three one-dimensional analytical problems that instantiate the bimodal setting in Fig. 2, where inputs are already compact and random anchor retrieval suffices. In each problem, the expert conditional is an equal mixture of two analytical modes. This setting lets us quantify directly whether the predictor can extrapolate both modes beyond the training support. Each method is trained on $20{,}000$ input–output pairs with three seeds and evaluated from 500 samples per input. NW_2 measures overall mismatch to the expert conditional, normalized by the distance between its modes, while balance error, measures deviation from the prescribed $50/50$ mode mass; formal definitions are given in Appx. E.

Method	NW_2 ID ↓	NW_2 OOD ↓	Balance OOD ↓
Oracle	0.078	0.086	0.019
FLOW	**0.221**	0.506	0.160
BRP	0.726	0.763	0.500
BFP	0.224	**0.321**	**0.103**

Table 1: **BFP distributionally extrapolates.** Results averaged over three problems and three seeds. Lower is better for all metrics. *Oracle* samples directly from the expert conditional and gives the Monte Carlo floor.

Hypothesis. We expect FLOW to preserve multimodality but extrapolate it unreliably, and deterministic BRP to extrapolate structurally but fail to represent both modes.

Result. BFP and FLOW show similar ID NW_2, but BFP reduces OOD NW_2 from 0.506 to 0.321 (36.6%) and improves mode balance from 0.160 to 0.103. BRP has the highest OOD distribution error and the maximal balance error of 0.500, reflecting its inability to represent both modes with deterministic regression. Thus, when representation and retrieval are controlled, only BFP extrapolates the full bimodal conditional distribution. Per-problem results are provided in Appx. E.

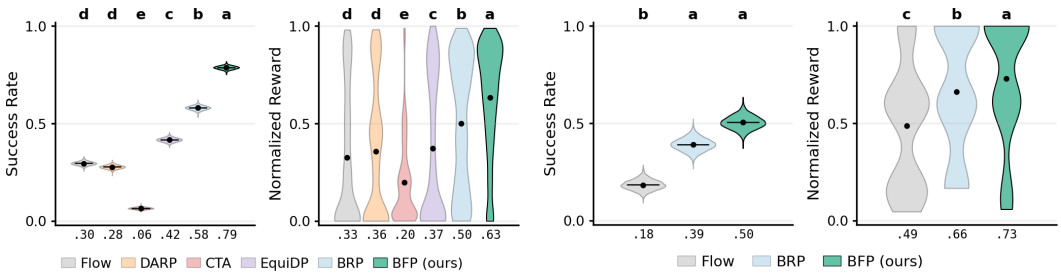
5.2 DOES BFP IMPROVE UNSEEN GOAL GENERALIZATION?

Settings. We evaluate seven simulation settings spanning the five manipulation tasks shown in Fig. 11, together with two real-world tasks in Fig. 6. Exact demonstration, ID, and OOD supports are provided in Appxs. A and B. We report results under state-domain and visuomotor observations. State uses simulator states. Visuomotor uses frozen R3M features (Nair et al., 2022), proprioception, and the explicit goal. We study other pretrained visual features (e.g., DINO-v3) in Sec. 5.4.

We compare the following: FLOW (Lipman et al., 2024), a standard generative GCIL policy, and four extrapolation-targeted baselines. EQUIDP (Wang et al., 2024a) represents prespecified extrapolation by encoding a known geometric equivariance into a generative policy. DARP (Pfeifer et al., 2026) retrieves the k nearest anchors and predicts from each anchor and its residual using a local, non-bilinear model. BRP (Netanyahu et al., 2023) uses a hand-crafted time-synchronized anchor policy and deterministic low-rank bilinear regression; for visuomotor tasks, we equip BRP with the same frozen pretrained visual representation as BFP. CTA (Kim et al., 2026) learns a task-endogenous analogy and uses it as the residual in hierarchical bilinear transduction; a high-level policy predicts an analogy, and a low-level policy maps it to primitive actions. See Appx. F for details.

For each pair of task and observation space in simulation, every training seed is evaluated on fixed 100 ID and 100 OOD initial conditions. We train 3 training seeds and report the mean and standard error of the mean (SEM) across the seeds. We use statistical hypothesis testing for policy comparison with 5% family-wise error rate (FWER) control within each task (Snyder et al., 2025; 2026).

Simulation Result. As shown in Fig. 4a, BFP achieves the highest OOD success on all seven settings, averaging 79%, compared with 58% for BRP and 30% for FLOW. The normalized dense-



(a) OOD performance in simulation.

(b) OOD performance in real-world.

Figure 4: **BFP generalizes in visuomotor manipulation.** **(a)** Success rate / Normalized reward averaged across seven simulation tasks **(b)** Same across hardware tasks. Success rate violins show Bayesian posterior estimates under a uniform Beta prior. Normalized reward violins illustrate empirical distributions of the results. Dots represent sample means and horizontal lines show posterior means (success-rate only). Letters on top of the plots show statistical comparison, where columns not sharing the same alphabet are significantly different with 95% confidence.

reward analysis shows the same ordering, with BFP forming the top FWER-controlled statistical group as indicated by the Compact Letter Display (CLD) (Piepho, 2004) shown on top of each subplot. Columns not sharing the same alphabet are significantly different with 5% FWER, verifying that BFP outperforms all the baselines. ID success remains high at 97%, indicating that the OOD gains do not come from sacrificing performance within demonstration support. Same trend holds for state-space result. Detailed analysis and per-task results are provided in Appx. B.

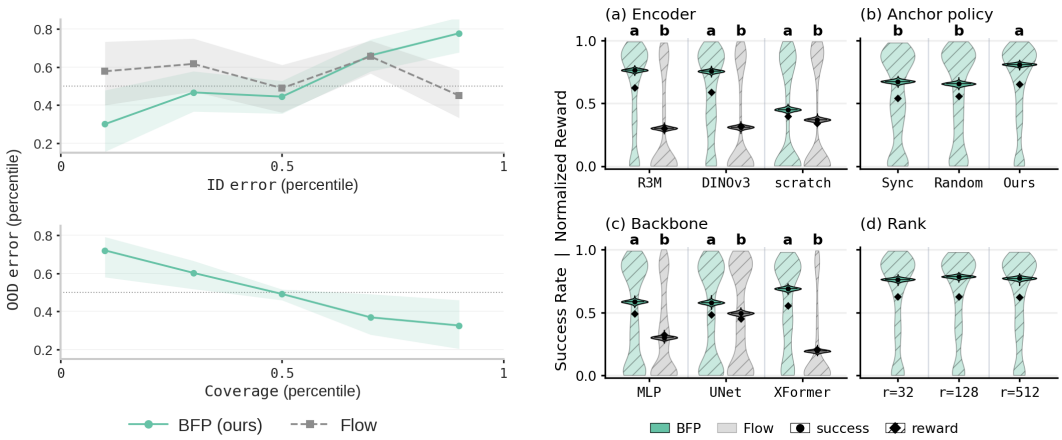
The baselines clarify where the gain comes from. BFP outperforms EQUIDP despite its carefully tuned (see Appx. F), task-specific geometric rule, showing the benefit of learning the extrapolation relation from demonstrations rather than prescribing it *a priori*. Its gains over DARP and CTA show that local retrieval or learned analogy alone does not match the proposed structured predictor, while the 21% gain over BRP indicates that learned anchor retrieval and generative bilinear-flow prediction add to the fixed-anchor deterministic bilinear recipe.

Does extrapolation transfer to the real-world? We evaluate generalization to unseen goals on a real-world robot in a pick-and-place task (`PnPMug`) and a pipette insertion task (`InsertPipette`) the latter requiring within 2cm precision. As shown in Fig. 4b, empirical results in this setting show the same trends observed in simulation. BFP achieves 50% average OOD success rate across the two tasks, compared with 18% for FLOW and 39% for BRP. These results suggest that BFP's generalization to unseen goals through extrapolation holds under real-world conditions including perception and goal estimation error. We report experimental setup and discuss results further in Appx. A.

5.3 IN-DISTRIBUTION DIAGNOSTICS OF EXTRAPOLATION

Finally, we ask whether the conditions in our extrapolation bound (Thm. 1) can be quantified and used as diagnostics before closed-loop OOD rollout. The bound depends on two quantities: in-distribution flow error $\mathcal{E}_{\text{tr}}^{\text{flow}}(\theta)$ to represent **(S) Bilinear Structure** and coverage score of residual (Kynkäänniemi et al., 2019) to represent **(C) Residual Coverage**. These yield two practical questions: can held-out ID flow error identify which trained checkpoint will extrapolate better, and, for a fixed checkpoint, can residual coverage identify which unseen goal-conditioned queries are better supported? For the first probe, we rank checkpoints within matched task–encoder–retrieval pools by held-out ID flow error and compare their normalized OOD flow-error ranks, where lower is better. For the second, we group OOD queries within each checkpoint by residual-coverage decile and compare their normalized flow error. Exact probe construction is deferred to Appx. F.3.

Vanilla FLOW shows the failure first: its OOD-error percentile does not rise with ID error. Thus, a checkpoint that fits held-out demonstrations better is not systematically a better OOD checkpoint. For BFP, the higher ID error worsens the normalized OOD-error rank by 0.48, i.e., 48 percentile points. Held-out ID flow error can therefore guide checkpoint selection, and hyperparameter search for BFP without OOD rollout.



(a) Diagnostics. (b) Ablations.

Figure 5: **Pre-rollout diagnostics and BFP ablations. (a)** Held-out ID flow error ranks OOD error for BFP but not for FLOW; within a fixed BFP checkpoint, better residual coverage is associated with lower extrapolation error. **(b)** Ablations of the visual encoder, anchor policy, backbone, and bilinear rank. Frozen pretrained features and phase-conditioned retrieval both lead to better performance; BFP can exploit stronger sequence backbones and is insensitive to rank.

Residual coverage provides a complementary query-level signal. Within a fixed BFP checkpoint, queries whose retrieved residuals fall in the lowest-coverage decile incur $2.6\times$ the flow error of those in the highest-coverage quintile. Residual coverage therefore serves as a query-level predictor of performance on unseen goals and motivates test-time retrieval that favors anchors inducing residuals well covered by training.

5.4 ABLATIONS

We ablate the visual representation, anchor retrieval, flow predictor backbone, and bilinear rank in Fig. 5b. A frozen pretrained visual representation and phase-conditioned retrieval are the most consequential components: BFP with a frozen R3M or DINOv3 encoder reaches both 76% OOD success, $+31\%$ over jointly training the encoder from scratch (45%), and learned phase-conditioned retrieval reaches 81% versus 67% and 66% for time-synchronized (as in BT) and random retrieval.

A possible concern is that BFP extrapolates only because its low-rank structure underfits, and that a more expressive architecture would erase the gain by overfitting. The backbone sweep shows the opposite. Replacing the MLP with a sequence-aware Transformer raises BFP from 59% to 69% OOD success, while the same backbone change lowers Flow from 30% to 20%, despite 94–99% ID success across trained variants. Thus, BFP can exploit a stronger sequence-aware flow backbone; what constrains OOD behavior is not limited network capacity, but the bilinear interaction through which the backbone combines the anchor and residual. Finally, performance is insensitive to rank.

6 CONCLUSION

We introduce BFP, a generative goal-conditioned visuomotor policy that extrapolates to unseen goals by learning generalizable observation-goal-action relations among demonstrated conditions. BFP extends bilinear transduction to conditional flow matching and combines it with compact representations and learned anchor selection to generate multimodal action distributions beyond demonstration support. Across seven simulation manipulation tasks and two real-world tasks, BFP substantially improves OOD success over competing methods, while providing practical signals for checkpoint selection and for identifying unseen conditions where extrapolation is better supported.

Limitations and Future Work. BFP currently assumes compact goal representations. Extending BFP to image- or language-conditioned goals remains an important direction for future work.

AI USE STATEMENT

We used generative AI tools to assist with literature review, experiment analysis, implementation of baseline methods, and editing and organization of manuscript text. AI-assisted literature review was verified against the original sources, baseline implementations were reviewed and tested by the authors, and all AI-assisted writing was manually reviewed and revised. We take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We are committed to making our results reproducible. The main paper specifies the BFP architecture, and pseudocode details provided in the appendix Appx. D. The appendix also documents the real-world task setups Appx. A. We release the implementation, experiment configurations, and supporting materials through our project website: `https://bilinearflow.netlify.app`.

REFERENCES

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

Yiming Ding, Carlos Florensa, Pieter Abbeel, and Mariano Phielipp. Goal-conditioned imitation learning. *Advances in neural information processing systems*, 32, 2019.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1801–1810. IEEE, 2019.

Niklas Funk, Julen Urain, Joao Carvalho, Vignesh Prasad, Georgia Chalvatzaki, and Jan Peters. Actionflow: Equivariant, accurate, and efficient policies with spatially symmetric flow matching. *arXiv preprint arXiv:2409.04576*, 2024.

Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *arXiv preprint arXiv:1910.11956*, 2019.

Zhang-Wei Hong, Ge Yang, and Pulkit Agrawal. Bilinear value networks. *arXiv preprint arXiv:2204.13695*, 2022.

Junseok Kim, Dohyeong Kim, Mineui Hong, and Songhwai Oh. Compositional transduction with latent analogies for offline goal-conditioned reinforcement learning. *arXiv preprint arXiv:2605.20609*, 2026.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019.

Daniel Lawson, Adriana Hugessen, Charlotte Cloutier, Glen Berseth, and Khimya Khetarpal. Self-predictive representations for combinatorial generalization in behavioral cloning. In *International Conference on Learning Representations*, volume 2026, pp. 101559–101586, 2026.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow Matching Guide and Code, December 2024. URL `http://arxiv.org/abs/2412.06264`. arXiv:2412.06264 [cs].

Yunhao Luo, Utkarsh Mishra, Yilun Du, and Danfei Xu. Generative trajectory stitching through diffusion composition. *Advances in Neural Information Processing Systems*, 38:37809–37843, 2026.

Lucas Manuelli, Wei Gao, Peter Florence, and Russ Tedrake. kpam: Keypoint affordances for category-level robotic manipulation. In *The International Symposium of Robotics Research*, pp. 132–157. Springer, 2019.

Marius Memmel, Jacob Berg, Bingqing Chen, Abhishek Gupta, and Jonathan Francis. Strap: Robot sub-trajectory retrieval for augmented policy learning. In *International Conference on Learning Representations*, volume 2025, pp. 63813–63832, 2025.

Utkarsh Aashu Mishra, Shangjie Xue, Yongxin Chen, and Danfei Xu. Generative skill chaining: Long-horizon skill planning with diffusion models. In *Conference on Robot Learning*, pp. 2905–2925. PMLR, 2023.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

Aviv Netanyahu, Abhishek Gupta, Max Simchowitz, Kaiqing Zhang, and Pulkit Agrawal. Learning to extrapolate: A transductive approach. *arXiv preprint arXiv:2304.14329*, 2023.

Adam Ousherovitch and Yixin Wang. Learning to extrapolate to new tasks: A relational approach to task extrapolation. In *Forty-third International Conference on Machine Learning*, 2026.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations*, volume 2025, pp. 94937–94982, 2025.

Gabriel Peyré and Marco Cuturi. *Computational optimal transport: With applications to data science*. Now Foundations and Trends, 2019.

Quinn Pfeifer, Ethan Pronovost, Paarth Shah, Khimya Khetarpal, Siddhartha Srinivasa, and Abhishek Gupta. Difference-aware retrieval policies for imitation learning. *arXiv preprint arXiv:2606.09758*, 2026.

Hans-Peter Piepho. An algorithm for a letter-based representation of all-pairwise comparisons. *Journal of Computational and Graphical Statistics*, 13(2):456–466, 2004.

Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. *arXiv preprint arXiv:2304.02532*, 2023.

Moritz Reuss, Ömer Erdinç Yağmurlu, Fabian Wenzel, and Rudolf Lioutikov. Multimodal diffusion transformer: Learning versatile behavior from multimodal goals. *arXiv preprint arXiv:2407.05996*, 2024.

Devavrat Shah, Dogyoon Song, Zhi Xu, and Yuzhe Yang. Sample efficient reinforcement learning via low-rank matrix estimation. *Advances in Neural Information Processing Systems*, 33:12092–12103, 2020.

Max Simchowitz, Abhishek Gupta, and Kaiqing Zhang. Tackling combinatorial distribution shift: A matrix completion perspective. In *The Thirty Sixth Annual Conference on Learning Theory*, pp. 3356–3468. PMLR, 2023.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

David Snyder, Asher James Hancock, Apurva Badithela, Emma Dixon, Patrick Miller, Rares Andrei Ambrus, Anirudha Majumdar, Masha Itkina, and Haruki Nishimura. Is your imitation learning policy better than mine? policy comparison with near-optimal stopping. *arXiv preprint arXiv:2503.10966*, 2025.

David Snyder, Apurva Badithela, Nikolai Matni, George Pappas, Anirudha Majumdar, Masha Itkina, and Haruki Nishimura. Beyond binary success: Sample-efficient and statistically rigorous robot policy comparison. *arXiv preprint arXiv:2603.13616*, 2026.

Julen Urain, Niklas Funk, Jan Peters, and Georgia Chalvatzaki. Se (3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. In *2023 IEEE international conference on robotics and automation (ICRA)*, pp. 5923–5930. IEEE, 2023.

Julen Urain, Ajay Mandlekar, Yilun Du, Mahi Shafiullah, Danfei Xu, Katerina Fragkiadaki, Georgia Chalvatzaki, and Jan Peters. Deep generative models in robotics: A survey on learning from multimodal demonstrations. *arXiv preprint arXiv:2408.04380*, 2024.

Dian Wang, Stephen Hart, David Surovik, Tarik Kelestemur, Haojie Huang, Haibo Zhao, Mark Yeatman, Jiuguang Wang, Robin Walters, and Robert Platt. Equivariant diffusion policy. *arXiv preprint arXiv:2407.01812*, 2024a.

Lirui Wang, Jialiang Zhao, Yilun Du, Edward H Adelson, and Russ Tedrake. Poco: Policy composition from and for heterogeneous robot learning. *arXiv preprint arXiv:2402.02511*, 2024b.

Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17868–17879. IEEE, 2024.

Rui Yang, Lin Yong, Xiaoteng Ma, Hao Hu, Chongjie Zhang, and Tong Zhang. What is essential for unseen goal generalization of offline goal-conditioned rl? In *International Conference on Machine Learning*, pp. 39543–39571. PMLR, 2023.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

Yifeng Zhu, Zhenyu Jiang, Peter Stone, and Yuke Zhu. Learning generalizable manipulation policies with object-centric 3d representations. *arXiv preprint arXiv:2310.14386*, 2023a.

Yifeng Zhu, Abhishek Joshi, Peter Stone, and Yuke Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors. In *Conference on Robot Learning*, pp. 1199–1210. PMLR, 2023b.

APPENDIX

We include several appendices with additional details, derivations, and results:

- Appx. A describes the real-world robot setup, tasks, and additional hardware results.
- Appx. B details the simulation benchmark, including tasks, goal splits, evaluation protocol, and additional results.
- Appx. C develops the theory: Appx. C.1 reviews the prior low-rank bilinear extrapolation result, Appx. C.2 shows that BFP conditional flow matching induces exactly such a bivariate regression problem, and Appx. C.3 proves the distributional extrapolation guarantee and its multi-anchor corollary.
- Appx. D gives the train- and test-time anchor-policy instantiations and their training and inference algorithms.
- Appx. E reports a controlled numerical study of distributional extrapolation.
- Appx. F describes how each baseline is instantiated, reports encoder and baseline sensitivity studies, and evaluates the pre-rollout extrapolation diagnostics.

A Real-world Experiment Details & Analysis

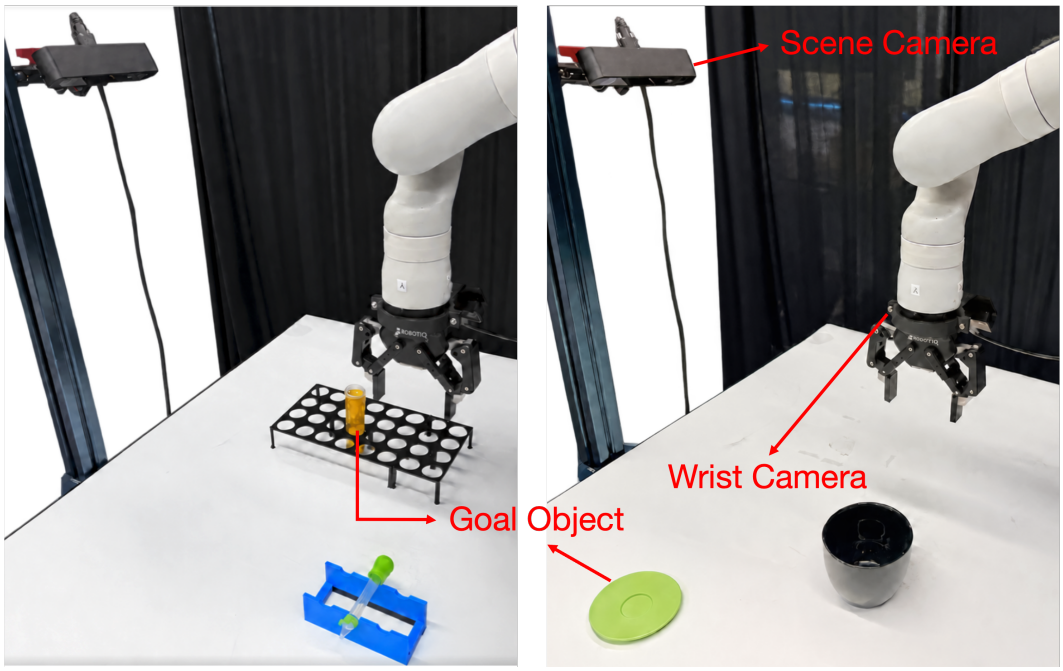


Figure 6: **Task Setup. (Left)** `InsertPipette` **(Right)** `PnPMug`

Hardware Setup. We conduct all real-world experiments on a 7-DoF Kinova Gen3 arm with a Robotiq 2F-85 parallel-jaw gripper; the full setup is shown in Fig. 6. We use two RGB-D cameras: a ZED 2i as a scene camera, and an OAK-D Pro as a wrist camera. Depth information is only used to extract the goal and not used for policy learning. The controller operates at 20 Hz. Inference runs on a single NVIDIA RTX 4090 Laptop GPU with 16 GB of memory. We use a Meta Quest 3 VR headset to collect teleoperated demonstrations.

The explicit task goal is extracted from the ZED RGB-D stream and represented in the robot base frame using hand–eye calibration. SAM 3 (Carion et al., 2026) segments the target plate and FoundationPose (Wen et al., 2024) registers and tracks its 6-DoF pose.

Task and Distribution Shift. We evaluate two goal-conditioned manipulation tasks, shown in Fig. 6. In `PnPMug`, the robot picks up a black mug and places it on a green plate. We use the 3D position of the plate as the policy goal. We collect 127 teleoperated demonstrations for policy training with the plate restricted to $y > 0$ and $x < 0.7$. For OOD evaluation, the plate is placed exclusively in the held-out region, such that no goal lies within the bounding box of the demonstrated goals. We evaluate 50 OOD initial conditions (ICs) with 2 trials per IC and 50 ID initial conditions with 1 trial per IC, with a maximum episode length of 30 s (1.5× average human demo duration).

`InsertPipette` evaluates more precise goal extrapolation. We use the discrete coordinate of test tube as the policy goal. The robot picks up a pipette from a cradle and inserts it into a test tube positioned in one of the holes of a 4×8 rack, whose hole pitch is 36 mm. The demonstration set for policy training contains 133 trajectories spanning 10 training holes in a 2×5 region. OOD evaluation uses the remaining 22 held-out holes, with five trials per hole. Each policy is evaluated in-distribution with 3 trials per hole. Maximum episode length is set to 60 s (1.5× average human demo duration).

Policy Training. All policies use the same frozen, pretrained R3M ResNet-34 visual encoder (Nair et al., 2022). The policy input consists of both camera views, a 10-dimensional proprioceptive state containing the end-effector position, 6-D rotation representation, and gripper state, together with the explicit task goal. Policies output absolute end-effector poses and gripper commands as 16-step action chunks using a 1-D U-Net backbone. We use prediction horizon and execution horizon $(T_p, T_a) = (16, 8)$. We train all policies with a batch size of 32 and a learning rate of 10^{-4}. Flow-

Table 2: Performance in- and out-of-distribution on real-world robot hardware. Success rate (SR, %), our primary metric, is highlighted in blue. Reward denotes normalized dense reward ($\times 100$). Higher is better for both metrics. Values are means $\pm$ standard error over trials: `PnPMug` 50 ID / 100 OOD trials, `InsertPipette` 30 ID / 110 OOD trials per method. Bold and underlined values indicate the best and second-best means, respectively, including ties.

Method	Metric	`PnPMug`	`InsertPipette`	Avg.
(a) In-distribution (ID)				
FLOW	**SR**	86 ± 5	83 ± 7	85
	Reward	95 ± 2	$\underline{92 \pm 3}$	93
BRP	**SR**	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$	**100**
	Reward	$\underline{97 \pm 1}$	$\mathbf{99 \pm 1}$	$\underline{98}$
BFP	**SR**	$\underline{98 \pm 2}$	$\underline{97 \pm 3}$	$\underline{97}$
	Reward	$\mathbf{99 \pm 1}$	$\mathbf{99 \pm 1}$	**99**
(b) Out-of-distribution (OOD)				
FLOW	**SR**	18 ± 4	18 ± 4	18
	Reward	35 ± 3	61 ± 2	48
BRP	**SR**	$\underline{45 \pm 5}$	$\underline{34 \pm 5}$	$\underline{39}$
	Reward	$\underline{62 \pm 4}$	$\underline{70 \pm 2}$	$\underline{66}$
BFP	**SR**	$\mathbf{51 \pm 5}$	$\mathbf{50 \pm 5}$	**50**
	Reward	$\mathbf{66 \pm 4}$	$\mathbf{79 \pm 2}$	**73**

based policies use 10 integration steps at inference. All hardware results use a single trained seed per method.

Baselines and Evaluation. We compare BFP with two baselines. FLOW (Lipman et al., 2024) is a goal-conditioned flow-matching policy that uses the same visual encoder and action-generation backbone as BFP, but removes the anchor–residual decomposition and bilinear interaction. BRP uses bilinear action regression, but replaces our learned candidate-set anchor scorer with synchronized, phase-based anchor selection. We use temporal ensemble technique (Zhao et al., 2023)

A human evaluator determines binary task success. We additionally report a normalized dense reward in $[0, 1]$ based on sequential task progress. For `PnPMug`, the progress sequence is *reach* $\rightarrow$ *grasp* $\rightarrow$ *lift* $\rightarrow$ *release* $\rightarrow$ *place on plate* $\rightarrow$ *retreat*. For `InsertPipette`, it is *reach* $\rightarrow$ *grasp-and-lift* $\rightarrow$ *align* $\rightarrow$ *insert* $\rightarrow$ *retreat*, where alignment requires the pipette to come within 2 cm of the target hole. A stage contributes to the reward only after all preceding stages have been completed, while partial progress toward the next stage is credited fractionally.

Results. Tab. 2 reports the complete task-level OOD and ID results. We report both success rate and normalized dense reward; the corresponding task-averaged comparison is discussed in the main text. Figs. 8 and 9 visualize the full per-task spatial distributions of successes and failures for both tasks. Fig. 10 visualizes the per-task distribution of dense reward for both tasks. Finally, Fig. 7 shows time-lapse strips with progress phase labels on successful BFP rollouts for both tasks. This figure also shows failed BRP and FLOW rollouts which exhibit the common failure modes of these baselines.



Figure 7: **OOD Rollout Strips.** Successful BFP rollouts in OOD labeled with the progress phase used to compute dense reward. BRP and FLOW failure rollouts in OOD exhibit common failure modes. In `PnPMug` and `InsertPipette`, FLOW often fails early in the task in grasping the manipuland despite its unchanging pose in the scene between ID training demonstrations and OOD rollouts. BRP's time-synchronized anchor cannot re-synchronize once the robot lags behind it, so the residual drifts off support and leads to a freezing failure mode where the policy fails to make further progress. In `PnPMug` this often occurs in the grasping phase, and in `InsertPipette` often occurs during the alignment phase.

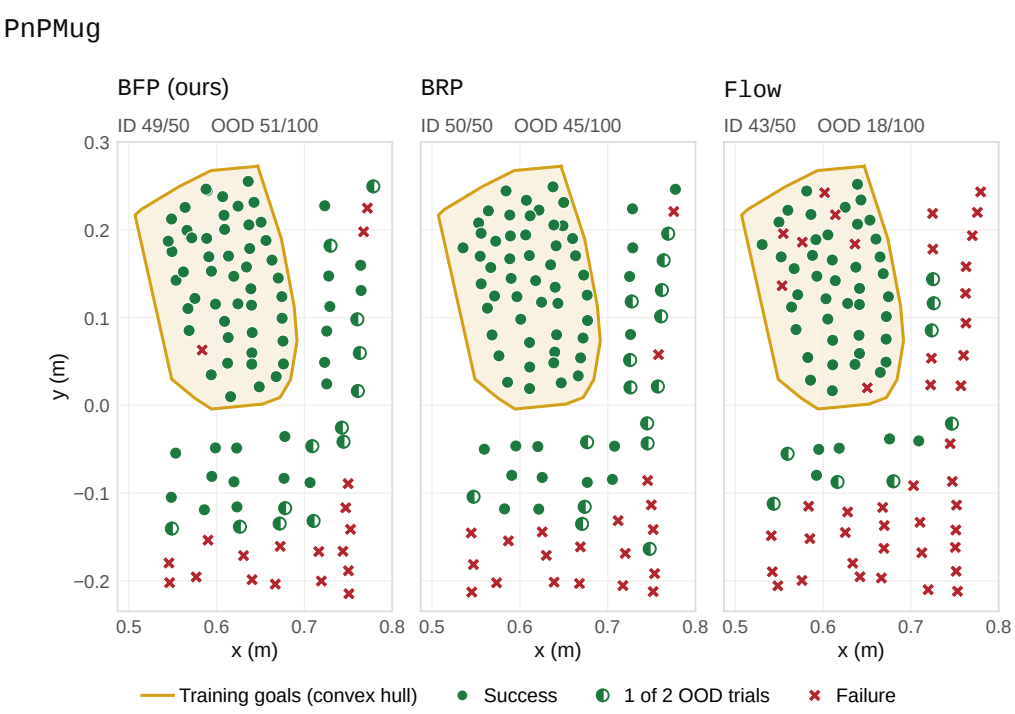


Figure 8: **`PnPMug` spatial distribution of successes and failures**. Each dot represents a goal object (plate) IC used in evaluation. Failures increase as the goal object is moved further from the training data distribution (ID). BFP achieves successful generalization to these OOD conditions more often and further away from the training data goal pose distribution compared to FLOW.

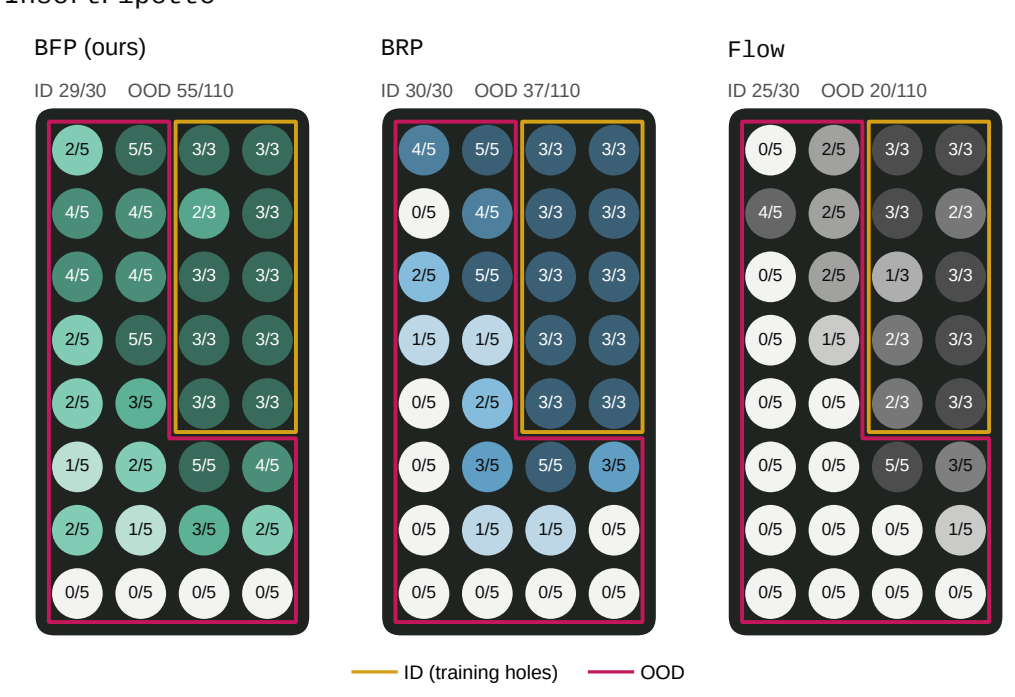


Figure 9: **`InsertPipette` spatial distribution of successes and failures**. Each circle represents a test-tube IC used for evaluation in a 4×8 test-tube tray. Generalization to unseen, OOD goals is substantially improved with BFP.

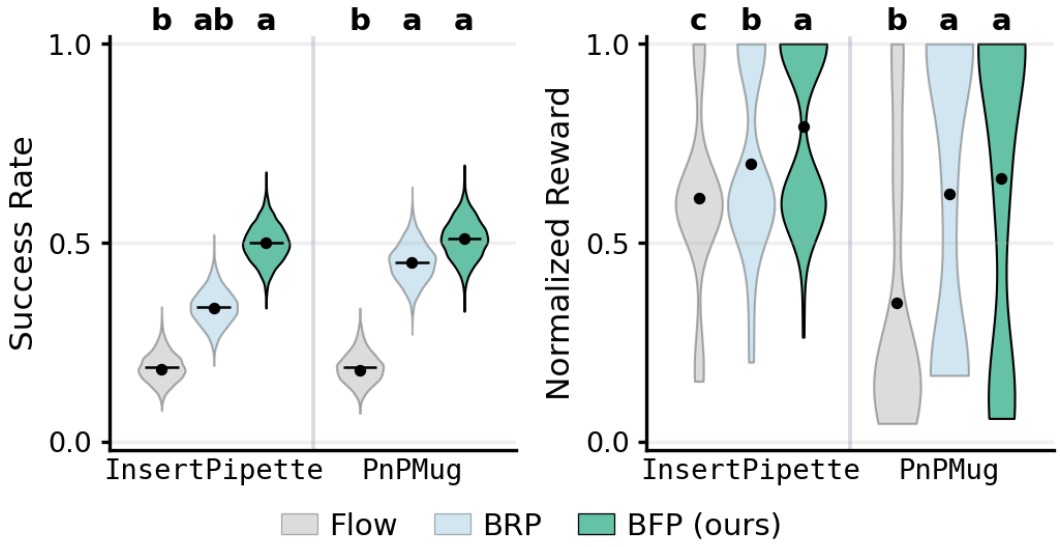


Figure 10: **Success rate and Dense reward under OOD evaluation on real-world robot hardware.**

B Simulation Benchmark Details and Additional Results

Tasks and goals. We evaluate five manipulation tasks across seven settings: `PushCube-D0/D1`, `Relocate-D0/D1`, `RotatePen`, `StackCube`, and `PickCube` (Fig. 11). The tasks span versatile skills including planar pushing, dexterous relocation and rotation, pick-and-place, and stacking. Each policy is conditioned on an explicit task goal; the goal definition, success criterion, and dense reward for each task are summarized in Tab. 3.

Distribution shifts. We evaluate extrapolation by holding out goal conditions from the training support (Fig. 12). The D0 settings isolate goal extrapolation under a restricted initial-condition distribution, whereas D1 evaluates the same goal shift with increased initial-condition variation. For tasks without a D0/D1 suffix, we use a D1 setting. Exact demonstration counts and train, ID, and OOD supports are reported in Tab. 4.

Training Details. All methods use the same demonstration set for each task and observation regime. 5% of demonstrations is held out for validation and is not used for policy training, as specified in Tab. 4. Method-specific architectures, extrapolation rules, and optimization details are provided in Appx. F.1.

Evaluation protocol and metrics. For each task–observation pair, each of three independently trained seeds is evaluated on the same 100 ID and 100 OOD initial conditions, yielding 200 evaluation rollouts per seed. We report the mean and standard error of the mean (SEM) across training seeds. Our primary metric is binary task success. We additionally report normalized dense reward, which measures partial task progress when the binary success criterion is not reached.

Results. Tab. 6 reports the complete task-level OOD results under state and visuomotor observations. We report both success rate and normalized dense reward; the corresponding task-averaged comparison is discussed in the main text. Figs. 13 and 14 visualize the full per-task distributions.

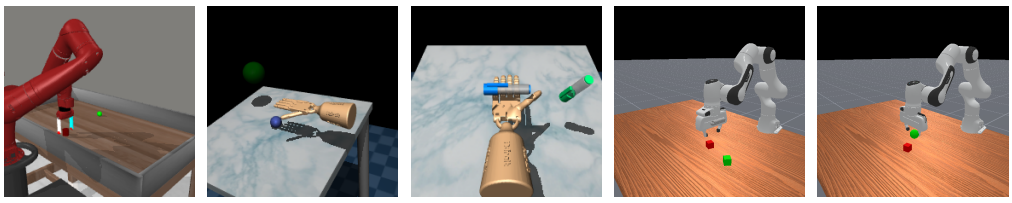
Tab. 5 reports the corresponding ID results. Across methods, ID performance is generally high relative to OOD performance, indicating that the primary differences between extrapolation methods emerge outside the demonstrated goal support. Figs. 15 and 16 provide the complete per-task ID distributions.

Table 3: **Simulation task definitions.** Goal denotes the compact representation used for policy conditioning. Success is evaluated at the final step of the episode. Dense reward is the environment's task-progress reward, normalized by the maximum episode return for each task.

Task (Benchmark)	Goal	Success criterion	Dense reward
`PushCube` (MetaWorld)	2D target position	target dist. < 5 cm	reach, grasp, and target progress
`Relocate` (Adroit)	3D target position	target dist. < 10 cm	reach, lift, and target progress
`RotatePen` (Adroit)	target orientation	ori. cos. > 0.95, pos. err. < 7.5 cm	orientation and position progress
`StackCube` (ManiSkill3)	target cube position	stacked, released, static	reach, grasp, stack, and release progress
`PickCube` (ManiSkill3)	3D target position	target dist. < 2.5 cm	reach, grasp, and place progress

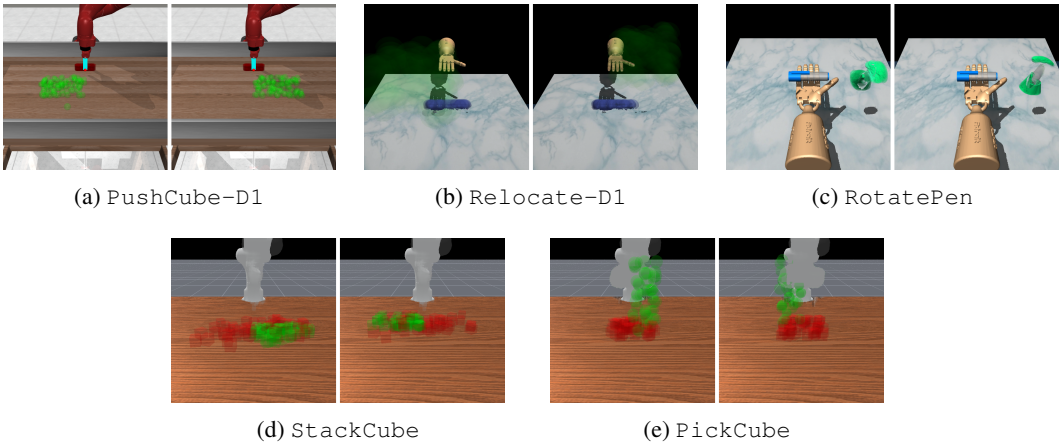
Table 4: **Simulation data and evaluation splits.** ID goals are sampled from the training goal support while OOD goals are held out. D0 fixes the object initial-state variation. Positions and angles are reported in meters and radians, respectively.

Setting	Demos	ID goal support	OOD goal support	Object initial condition
`PushCube-D0`	200	$x \in [0, 0.3]$ $y \in [0.5, 0.7]$	$x \in [-0.3, 0]$ $y \in [0.5, 0.7]$	fixed
`PushCube-D1`	200	$x \in [0, 0.3]$ $y \in [0.5, 0.7]$	$x \in [-0.3, 0]$ $y \in [0.5, 0.7]$	puck $x \in [-0.06, 0.06]$
`Relocate-D0`	100	$x, y \in [0, 0.3]$ $z \in [0.15, 0.35]$	$x, y \in [-0.3, 0]$ $z \in [0.15, 0.35]$	fixed
`Relocate-D1`	250	$x, y \in [0, 0.3]$ $z \in [0.15, 0.35]$	$x, y \in [-0.3, 0]$ $z \in [0.15, 0.35]$	ball $x \in [-0.1, 0.1]$
`RotatePen`	187	roll $\in [0.2, 1.0]$ pitch $\in [-0.2, 0.6]$	roll $\in [-0.3, 0]$ pitch $\in [-0.2, 0.6]$	fixed
`StackCube`	200	$x \in [0, 0.2]$ $y \in [0, 0.27]$	$x \in [-0.19, 0]$ $y \in [-0.25, 0]$	$x \in [-0.19, 0.19]$ $y \in [-0.29, 0.30]$
`PickCube`	200	$x, y \in (0, 0.1]$	$x, y \in [-0.1, 0)$	cube $x, y \in [-0.1, 0.1]$



(a) `PushCube` (b) `Relocate` (c) `RotatePen` (d) `StackCube` (e) `PickCube`

Figure 11: **Evaluation Tasks.** (a) `PushCube`, (b) `Relocate`, (c) `RotatePen`, (d) `StackCube`, (e) `PickCube`. The suite spans pushing, dexterous object relocation and rotation, pick-and-place, stacking, and insertion. For definition of ID/OOD per task, please see Fig. 12.



(a) `PushCube-D1` (b) `Relocate-D1` (c) `RotatePen`

(d) `StackCube` (e) `PickCube`

Figure 12: **Distribution Shift.** Each task panel visualizes the support of the training conditions (ID, left) and the held-out evaluation conditions (OOD, right). The paired views use the same camera and scene configuration so that the corresponding distribution shift is directly visible.

Table 5: In-distribution performance with state and visuomotor observations. Success rate (SR, %), our primary metric, is highlighted in blue. Reward denotes normalized dense reward ($\times 100$). Higher is better for both metrics. Bold and underlined values indicate the best and second-best means, respectively, including ties. Mean $\pm$ standard error over three training seeds.

Method	Metric	PushCube-D0	PushCube-D1	Relocate-D0	Relocate-D1	RotatePen	StackCube	PickCube	Avg.
(a) State									
FLOW	**SR**	**100 ± 0**	**100 ± 0**	98 ± 1	**100 ± 0**	97 ± 1	62 ± 3	96 ± 1	93
	Reward	**89 ± 0**	**88 ± 0**	81 ± 1	80 ± 1	87 ± 0	64 ± 0	63 ± 0	79
DARP	**SR**	**100 ± 0**	**100 ± 0**	**100 ± 0**	**100 ± 0**	97 ± 0	75 ± 0	87 ± 1	94
	Reward	**89 ± 0**	**88 ± 0**	**84 ± 1**	82 ± 1	87 ± 0	67 ± 0	64 ± 0	**80**
CTA	**SR**	**100 ± 0**	96 ± 3	**100 ± 0**	**100 ± 0**	97 ± 0	52 ± 2	82 ± 2	89
	Reward	**89 ± 0**	86 ± 2	78 ± 1	81 ± 1	87 ± 0	58 ± 0	62 ± 0	77
EQUIDP	**SR**	**100 ± 0**	**100 ± 0**	99 ± 0	**100 ± 0**	**98 ± 1**	58 ± 2	90 ± 1	92
	Reward	**89 ± 0**	**88 ± 0**	82 ± 1	**83 ± 1**	**88 ± 0**	63 ± 0	62 ± 0	79
BRP	**SR**	**100 ± 0**	**100 ± 0**	**100 ± 0**	99 ± 0	97 ± 0	74 ± 3	**100 ± 0**	96
	Reward	**89 ± 0**	**88 ± 0**	80 ± 0	81 ± 1	87 ± 0	66 ± 1	**67 ± 0**	**80**
BFP	**SR**	**100 ± 0**	**100 ± 0**	**100 ± 0**	**100 ± 0**	96 ± 0	**89 ± 1**	**100 ± 0**	**98**
	Reward	87 ± 0	**88 ± 0**	82 ± 1	81 ± 0	87 ± 0	**71 ± 1**	65 ± 0	**80**
(b) Visuomotor									
FLOW	**SR**	**100 ± 0**	94 ± 1	94 ± 2	95 ± 2	**99 ± 1**	60 ± 1	94 ± 1	91
	Reward	**89 ± 0**	83 ± 1	77 ± 1	79 ± 2	**89 ± 0**	60 ± 1	65 ± 0	77
DARP	**SR**	**100 ± 0**	97 ± 0	**100 ± 0**	89 ± 1	**99 ± 1**	77 ± 1	90 ± 1	93
	Reward	**89 ± 0**	86 ± 0	**81 ± 0**	75 ± 1	**89 ± 0**	68 ± 0	**68 ± 0**	79
CTA	**SR**	98 ± 0	75 ± 4	90 ± 2	74 ± 3	96 ± 2	55 ± 7	11 ± 0	71
	Reward	64 ± 8	43 ± 5	41 ± 1	36 ± 4	85 ± 0	58 ± 3	38 ± 1	52
EQUIDP	**SR**	**100 ± 0**	**100 ± 0**	**100 ± 0**	**100 ± 0**	97 ± 0	83 ± 3	94 ± 1	96
	Reward	**89 ± 0**	**88 ± 0**	80 ± 1	**83 ± 1**	87 ± 0	69 ± 2	**68 ± 1**	**81**
BRP	**SR**	**100 ± 0**	97 ± 1	**100 ± 0**	**100 ± 0**	97 ± 0	83 ± 2	**97 ± 1**	96
	Reward	**89 ± 0**	86 ± 1	80 ± 0	80 ± 0	86 ± 0	68 ± 1	67 ± 0	79
BFP	**SR**	**100 ± 0**	99 ± 1	**100 ± 0**	99 ± 0	98 ± 0	**88 ± 0**	96 ± 1	**97**
	Reward	87 ± 0	87 ± 1	80 ± 1	81 ± 0	87 ± 0	**70 ± 0**	66 ± 1	80

Table 6: OOD performance with state and visuomotor observations. Success rate (SR, %), our primary metric, is highlighted in blue. Reward denotes normalized dense reward ($\times 100$). Higher is better for both metrics. Bold and underlined values indicate the best and second-best means, respectively, including ties.

Method	Metric	PushCube-D0	PushCube-D1	Relocate-D0	Relocate-D1	RotatePen	StackCube	PickCube	Avg.
(a) State									
FLOW	**SR**	31 ± 4	49 ± 2	26 ± 7	24 ± 2	52 ± 5	41 ± 1	80 ± 2	43
	Reward	27 ± 4	42 ± 1	18 ± 5	22 ± 2	49 ± 5	55 ± 0	50 ± 3	38
DARP	**SR**	$\underline{36 \pm 1}$	45 ± 1	43 ± 11	47 ± 8	75 ± 10	2 ± 1	5 ± 1	36
	Reward	31 ± 1	40 ± 1	29 ± 7	36 ± 6	65 ± 8	38 ± 1	40 ± 0	40
CTA	**SR**	$\underline{36 \pm 8}$	48 ± 9	15 ± 4	24 ± 4	77 ± 12	12 ± 3	25 ± 3	34
	Reward	49 ± 4	45 ± 7	10 ± 2	16 ± 1	70 ± 10	35 ± 2	45 ± 2	39
EQUIDP	**SR**	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$	27 ± 4	85 ± 2	$\underline{88 \pm 1}$	$\underline{61 \pm 2}$	85 ± 1	78
	Reward	$\mathbf{88 \pm 0}$	$\mathbf{88 \pm 0}$	22 ± 4	53 ± 2	76 ± 1	$\underline{60 \pm 2}$	59 ± 0	64
BRP	**SR**	$\mathbf{100 \pm 0}$	$\underline{94 \pm 2}$	$\underline{76 \pm 9}$	$\underline{99 \pm 1}$	86 ± 1	51 ± 1	$\underline{99 \pm 0}$	$\underline{86}$
	Reward	$\underline{85 \pm 0}$	77 ± 1	$\underline{47 \pm 3}$	$\underline{72 \pm 2}$	$\underline{78 \pm 1}$	57 ± 1	$\mathbf{62 \pm 0}$	$\underline{68}$
BFP	**SR**	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$	$\mathbf{100 \pm 0}$	$\mathbf{96 \pm 1}$	$\mathbf{78 \pm 9}$	$\mathbf{100 \pm 0}$	**96**
	Reward	83 ± 0	$\underline{83 \pm 0}$	$\mathbf{67 \pm 1}$	$\mathbf{81 \pm 1}$	$\mathbf{81 \pm 2}$	$\mathbf{68 \pm 3}$	$\underline{61 \pm 1}$	**75**
(b) Visuomotor									
FLOW	**SR**	28 ± 0	34 ± 1	21 ± 3	17 ± 2	50 ± 4	11 ± 3	45 ± 6	30
	Reward	26 ± 0	31 ± 1	13 ± 1	14 ± 2	68 ± 3	34 ± 1	43 ± 2	33
DARP	**SR**	31 ± 3	23 ± 1	33 ± 8	$\underline{32 \pm 12}$	$\underline{57 \pm 7}$	16 ± 1	0 ± 0	28
	Reward	28 ± 2	21 ± 1	20 ± 5	$\underline{24 \pm 8}$	$\underline{69 \pm 4}$	43 ± 1	45 ± 0	36
CTA	**SR**	11 ± 1	8 ± 2	1 ± 0	0 ± 0	23 ± 3	1 ± 0	0 ± 0	6
	Reward	24 ± 2	12 ± 2	3 ± 0	2 ± 0	29 ± 5	38 ± 2	31 ± 1	20
EQUIDP	**SR**	66 ± 11	$\underline{59 \pm 4}$	7 ± 2	26 ± 4	14 ± 3	$\underline{41 \pm 2}$	$\underline{79 \pm 2}$	42
	Reward	58 ± 10	$\underline{51 \pm 3}$	6 ± 0	17 ± 2	25 ± 3	$\underline{47 \pm 2}$	57 ± 2	37
BRP	**SR**	$\underline{99 \pm 1}$	57 ± 0	$\underline{57 \pm 3}$	29 ± 3	$\underline{57 \pm 1}$	28 ± 1	$\underline{79 \pm 2}$	$\underline{58}$
	Reward	$\mathbf{78 \pm 1}$	50 ± 0	$\underline{40 \pm 1}$	19 ± 2	60 ± 2	40 ± 0	$\mathbf{62 \pm 0}$	$\underline{50}$
BFP	**SR**	$\mathbf{100 \pm 0}$	$\mathbf{83 \pm 1}$	$\mathbf{99 \pm 1}$	$\mathbf{54 \pm 4}$	$\mathbf{78 \pm 5}$	$\mathbf{48 \pm 5}$	$\mathbf{88 \pm 2}$	**79**
	Reward	$\underline{74 \pm 1}$	$\mathbf{72 \pm 1}$	$\mathbf{68 \pm 2}$	$\mathbf{38 \pm 1}$	$\mathbf{80 \pm 1}$	$\mathbf{52 \pm 1}$	$\underline{60 \pm 0}$	**63**

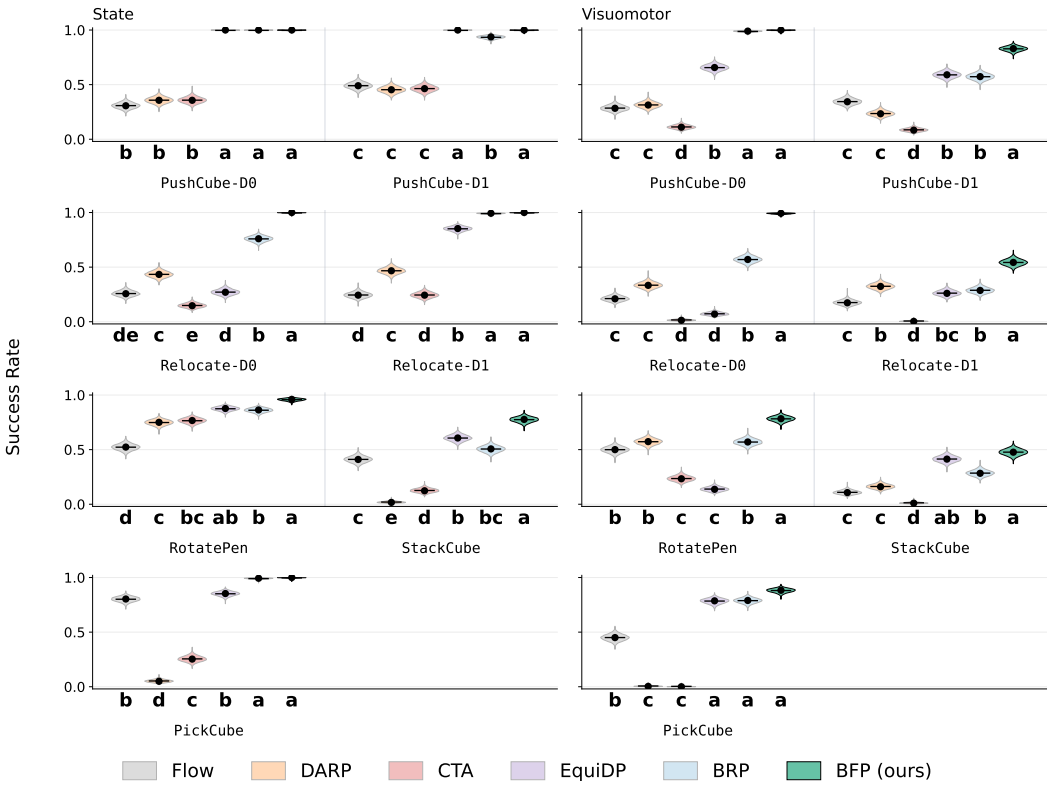


Figure 13: **Success rate under OOD evaluation.**

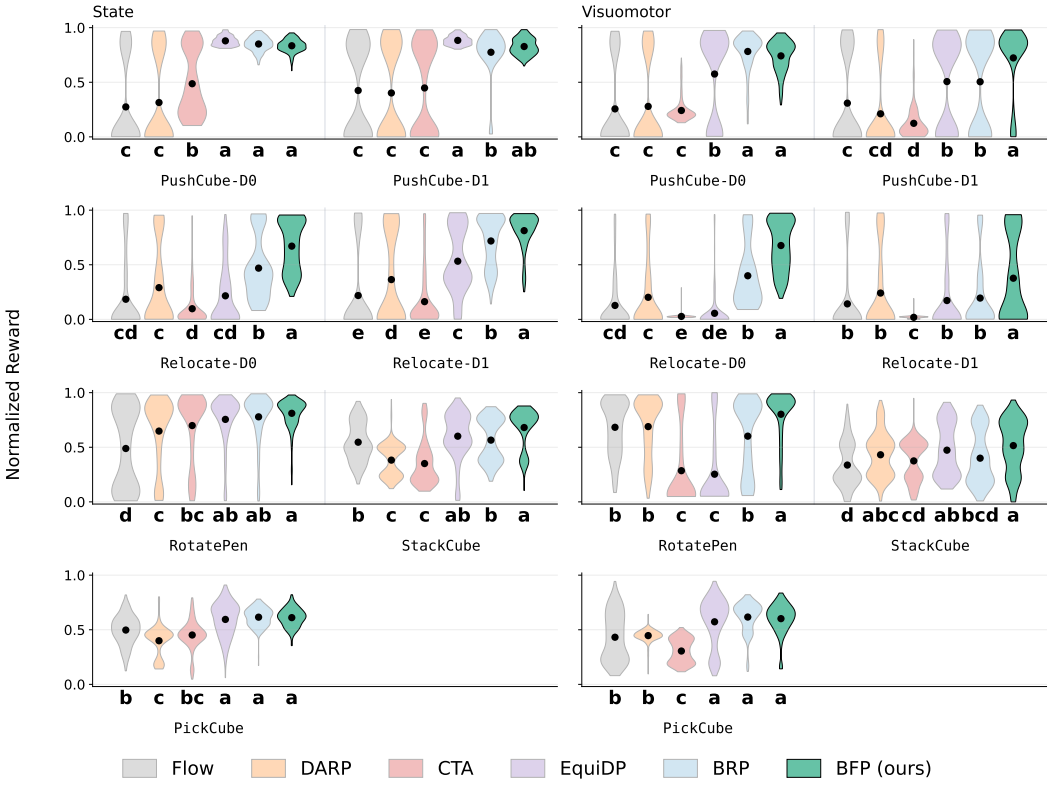


Figure 14: **Dense reward under OOD evaluation.**

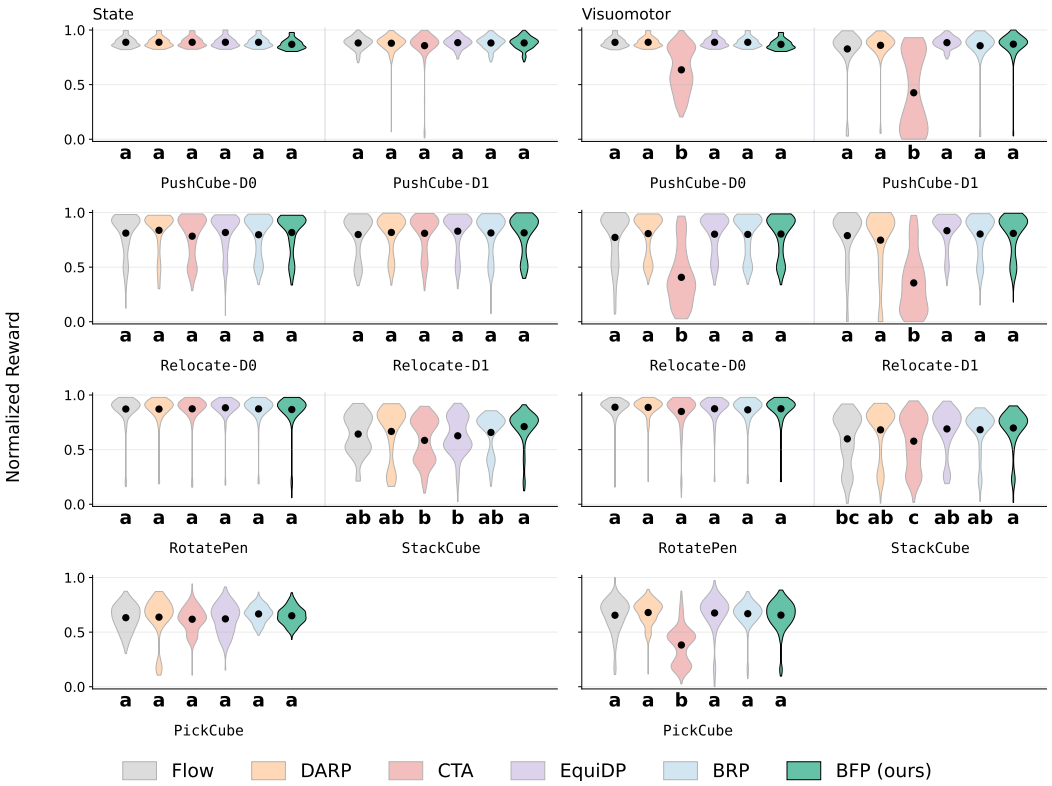


Figure 15: **Dense reward under ID evaluation.**

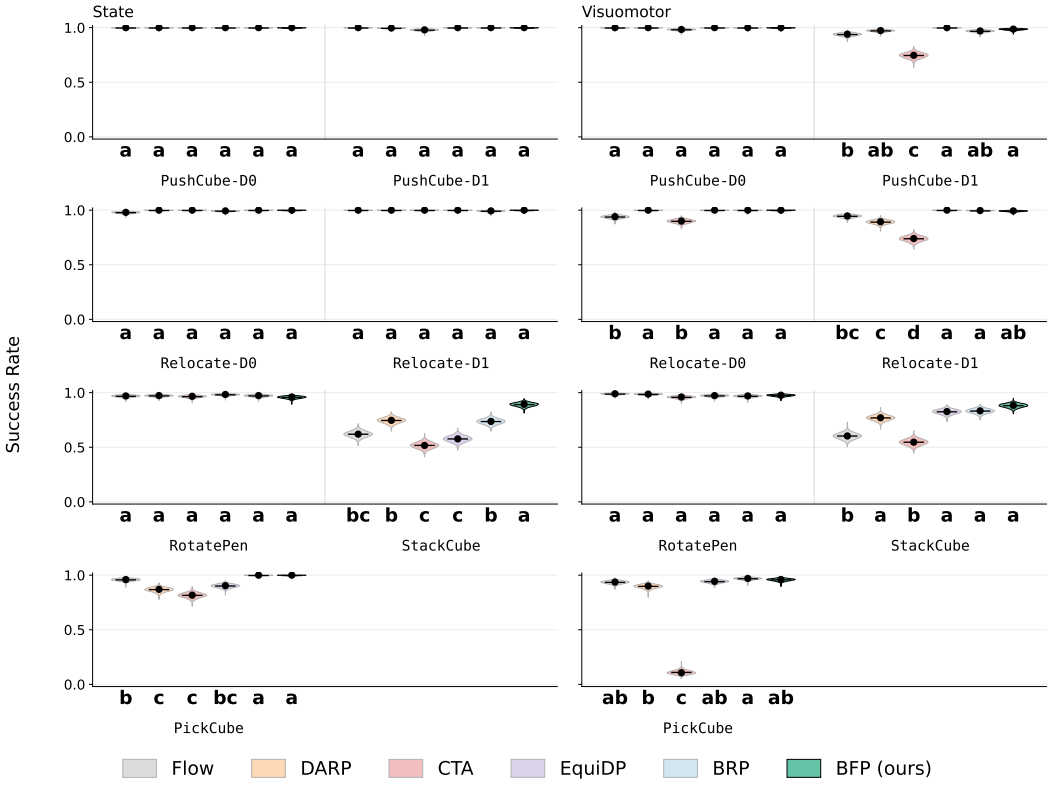


Figure 16: **Success rate under ID evaluation.**

C Theoretical Result

C.1 Background: Low-Rank Bilinear Combinatorial Extrapolation

In this section, we review the prior combinatorial-extrapolation result (Netanyahu et al., 2023; Simchowitz et al., 2023) used in the proof and analysis of our method. We first formulate the regression problem in the anchor–residual notation used by BFP, then state the low-rank bilinear extrapolation bound and its matrix-completion interpretation. Appx. C.2 shows how the BFP encoder, anchor policies, and conditional flow matching instantiate this problem for the velocity field.

C.1.1 Combinatorial extrapolation

Consider a regression problem in which a target Y is predicted from a residual feature Δ and an anchor feature $Z^\dagger$. For each regime $s \in \{\mathrm{tr}, \mathrm{te}\}$, let $\overline{D}_s$ denote the distribution of $(\Delta, Z^\dagger)$, and assume that training and test share the same Bayes regression function

$$h^\star(\boldsymbol{\delta}, \boldsymbol{z}^\dagger) := \mathbb{E}_s\left[Y \mid \Delta = \boldsymbol{\delta}, Z^\dagger = \boldsymbol{z}^\dagger\right], \qquad s \in \{\mathrm{tr}, \mathrm{te}\}. \tag{16}$$

A rank-r bilinear predictor has the form

$$h_\theta(\boldsymbol{\delta}, \boldsymbol{z}^\dagger) := \left\langle f_\theta(\boldsymbol{\delta}), g_\theta(\boldsymbol{z}^\dagger)\right\rangle_r, \qquad f_\theta(\boldsymbol{\delta}), g_\theta(\boldsymbol{z}^\dagger) \in \mathbb{R}^r. \tag{17}$$

Its squared prediction risk in regime s is

$$\mathcal{R}_s(h_\theta) := \mathbb{E}_{(\Delta, Z^\dagger)\sim\overline{D}_s}\left[\left(h_\theta(\Delta, Z^\dagger) - h^\star(\Delta, Z^\dagger)\right)^2\right]. \tag{18}$$

Combinatorial extrapolation asks when small training risk implies small test risk even though $\overline{D}_{\mathrm{te}}$ may contain residual–anchor combinations absent from $\overline{D}_{\mathrm{tr}}$. Vector-valued targets are handled coordinate-wise, and their squared risks are summed over output coordinates.

Notation. We use the BFP symbols Δ and $Z^\dagger$ from the outset. Within this appendix, they simply name the two regression features; in BFP, $Z^\dagger$ is the encoded anchor and Δ is the encoded query–anchor residual. All inner products $\langle\cdot,\cdot\rangle_r$ below are taken in $\mathbb{R}^r$.

Simchowitz et al. (2023) study this combinatorial-extrapolation problem directly for paired features. Bilinear Transduction (Netanyahu et al., 2023) instead constructs it from an ordinary regression problem: given a query X, select an anchor $X^\dagger$ and define

$$\Delta := X - X^\dagger, \qquad Z^\dagger := X^\dagger, \qquad h^\star(\boldsymbol{\delta}, \boldsymbol{z}^\dagger) := h^\star_{\mathrm{orig}}(\boldsymbol{z}^\dagger + \boldsymbol{\delta}). \tag{19}$$

BFP uses the same construction in representation space, $Z^\dagger = \phi(X^\dagger)$ and $\Delta = \phi(X) - \phi(X^\dagger)$; Appx. C.2 derives the resulting train and test laws and identifies the conditional-flow regression target.

C.1.2 Low-rank bilinear combinatorial extrapolation bound

For nonnegative finite measures μ and ν, write $\mu \preceq_\kappa \nu$ when

$$\mu(B) \le \kappa\nu(B) \qquad \text{for every measurable set } B. \tag{20}$$

The two assumptions below correspond to **(C) Residual Coverage** and **(S) Bilinear Structure** in the main text.

Assumption 2 (Combinatorial coverage)**.** *There exist residual marginals $D_{\Delta,1}, D_{\Delta,2}$, anchor marginals $D_{Z^\dagger,1}, D_{Z^\dagger,2}$, and a constant $\kappa \ge 1$ such that, with*

$$D_{i\otimes j} := D_{\Delta,i} \otimes D_{Z^\dagger,j}, \qquad i, j \in \{1, 2\}, \tag{21}$$

we have

$$\sum_{(i,j)\neq(2,2)} D_{i\otimes j} \preceq_\kappa \overline{D}_{\mathrm{tr}}, \qquad \overline{D}_{\mathrm{te}} \preceq_\kappa \sum_{i,j\in\{1,2\}} D_{i\otimes j}. \tag{22}$$

Assumption 2 requires training to cover the three blocks $D_{1\otimes1}$, $D_{1\otimes2}$, and $D_{2\otimes1}$, while evaluation may additionally contain the unseen combination $D_{2\otimes2}$.

Assumption 3 (Well-conditioned low-rank bilinear structure)**.** *There exist feature maps*

$$f^\star : \mathbb{Z}_\Delta \to \mathbb{R}^r, \qquad g^\star : \mathbb{Z}_\dagger \to \mathbb{R}^r, \tag{23}$$

such that, for all residual–anchor pairs,

$$h^\star(\boldsymbol{\delta}, \boldsymbol{z}^\dagger) = \left\langle f^\star(\boldsymbol{\delta}), g^\star(\boldsymbol{z}^\dagger) \right\rangle_r, \qquad \left|h^\star(\boldsymbol{\delta}, \boldsymbol{z}^\dagger)\right| \le M. \tag{24}$$

Furthermore, under $D_{1\otimes 1}$, define

$$\begin{aligned} \Sigma^\star_\Delta &:= \mathbb{E}_{\Delta\sim D_{\Delta,1}}\left[f^\star(\Delta) f^\star(\Delta)^\top\right], \\ \Sigma^\star_{Z^\dagger} &:= \mathbb{E}_{Z^\dagger\sim D_{Z^\dagger,1}}\left[g^\star(Z^\dagger) g^\star(Z^\dagger)^\top\right]. \end{aligned} \tag{25}$$

The feature directions are non-degenerate:

$$\sigma^2 := \lambda_{\min}(\Sigma^\star_\Delta)\lambda_{\min}(\Sigma^\star_{Z^\dagger}) > 0. \tag{26}$$

For BFP, this assumption becomes central design objective of encoder and retrieval policy. We aim to find a representation and retrieval policy that action is a low-rank bilinear product of residual and the anchor, while avoiding the collapse.

Theorem 4 (Low-rank bilinear combinatorial extrapolation bound; prior result)**.** *Suppose Assumption 2 and Assumption 3 hold, and let h_θ be the rank-r predictor in equation 17. If*

$$\mathcal{R}_{\mathrm{tr}}(h_\theta) \le \frac{\sigma^2}{4\kappa}, \tag{27}$$

then

$$\mathcal{R}_{\mathrm{te}}(h_\theta) \le \kappa^2\left(1 + \frac{64M^4}{\sigma^4}\right)\mathcal{R}_{\mathrm{tr}}(h_\theta). \tag{28}$$

C.1.3 Matrix-completion interpretation

A key proof idea underlying low-rank bilinear combinatorial extrapolation is its reduction to a structured missing-not-at-random matrix-completion problem (Shah et al., 2020; Simchowitz et al., 2023; Netanyahu et al., 2023). Suppose $D_{\Delta,i}$ and $D_{Z^\dagger,j}$ have probability masses $p_{\Delta,i}$ and $p_{Z^\dagger,j}$. For rank-r factors f and g, define the probability-weighted factor matrices

$$\begin{aligned} [\mathbf{F}_i]_{\boldsymbol{\delta},:} &:= \sqrt{p_{\Delta,i}(\boldsymbol{\delta})}\, f(\boldsymbol{\delta})^\top, \\ [\mathbf{G}_j]_{\boldsymbol{z}^\dagger,:} &:= \sqrt{p_{Z^\dagger,j}(\boldsymbol{z}^\dagger)}\, g(\boldsymbol{z}^\dagger)^\top. \end{aligned} \tag{29}$$

Let

$$\begin{aligned} \mathbf{M}_{i\otimes j}(f,g) &:= \mathbf{F}_i \mathbf{G}_j^\top, \\ [\mathbf{M}^\star_{i\otimes j}]_{\boldsymbol{\delta},\boldsymbol{z}^\dagger} &:= \sqrt{p_{\Delta,i}(\boldsymbol{\delta}) p_{Z^\dagger,j}(\boldsymbol{z}^\dagger)}\, h^\star(\boldsymbol{\delta}, \boldsymbol{z}^\dagger). \end{aligned} \tag{30}$$

Then, for every block (i,j),

$$\begin{aligned} &\mathbb{E}_{\substack{\Delta\sim D_{\Delta,i},\\ Z^\dagger\sim D_{Z^\dagger,j}}}\left[\left(\left\langle f(\Delta), g(Z^\dagger)\right\rangle_r - h^\star(\Delta, Z^\dagger)\right)^2\right] \\ &\qquad = \left\|\mathbf{M}_{i\otimes j}(f,g) - \mathbf{M}^\star_{i\otimes j}\right\|_F^2. \end{aligned} \tag{31}$$

Notice that, each product-block prediction risk is exactly a probability-weighted Frobenius error. BFP applies this trick to Flow Matching loss as a key proof idea to distributional extrapolation.

C.2 The Bivariate Conditional-Flow Problem Induced by BFP

This appendix shows how BFP constructs the bivariate regression problem via transductive lens of the core modules. We begin with the expert action distribution, re-express it in the anchor–residual coordinates induced by the BFP encoder and anchor policies, and then show that conditional flow matching turns action generation into regression of a marginal velocity field. After fixing a flow state, flow time, and output coordinate, this regression is exactly the combinatorial extrapolation problem of Appx. C.1.

C.2.1 BFP-INDUCED ANCHOR–RESIDUAL COORDINATES

Condition throughout on the realized demonstration dataset $\mathcal{D}$. For each regime $s \in \{\mathrm{tr}, \mathrm{te}\}$, let

$$X \sim P_s^X, \qquad \boldsymbol{A} \mid X = x \sim q^\star(\cdot \mid x), \tag{32}$$

where P_s^X is the marginal distribution of policy conditions at $s \in [\mathrm{tr}, \mathrm{te}]$ and $q^\star(\cdot \mid x)$ is the expert distribution over action chunks. Training and evaluation may use different condition distributions, but they share the same expert conditional $q^\star(\cdot \mid x)$. The distributional-extrapolation objective is to generate from this conditional when x is drawn from P_{te}^X, including conditions outside the support of P_{tr}^X.

For a query X, the anchor policy selects an anchor

$$X^\dagger \sim \rho_s(\cdot \mid X, \mathcal{D}), \qquad s \in \{\mathrm{tr}, \mathrm{te}\}. \tag{33}$$

BFP applies the shared observation encoder ϕ to the query and anchor and forms

$$\Xi := \phi(X), \qquad Z^\dagger := \phi(X^\dagger), \qquad \Delta := \Xi - Z^\dagger. \tag{34}$$

Thus, the encoder and anchor policy define the transformation

$$T_\phi(x, x^\dagger) := \big(\phi(x) - \phi(x^\dagger), \phi(x^\dagger)\big)\,. \tag{35}$$

Together with Eq. (32), this transformation induces a joint law

$$\overline{P}_s^{\phi,\rho_s} := \mathrm{Law}_s(\Delta, Z^\dagger, \boldsymbol{A}), \tag{36}$$

which is equivalently characterized by

$$\begin{aligned} &\mathbb{E}_{\overline{P}_s^{\phi,\rho_s}}\left[F(\Delta, Z^\dagger, \boldsymbol{A})\right] \\ &\quad = \mathbb{E}_{X\sim P_s^X}\mathbb{E}_{\boldsymbol{A}\sim q^\star(\cdot|X)}\mathbb{E}_{X^\dagger\sim\rho_s(\cdot|X,\mathcal{D})}\left[F\big(T_\phi(X, X^\dagger), \boldsymbol{A}\big)\right] \end{aligned} \tag{37}$$

for every bounded measurable F. Let $\overline{D}_s^{\phi,\rho_s}$ denote the $(\Delta, Z^\dagger)$ marginal of this law. This is the train or test combinatorial-extrapolation distribution constructed by the BFP modules.

The pair reconstructs the encoded query,

$$\Xi = \Delta + Z^\dagger, \tag{38}$$

but it need not reconstruct the raw condition X. We therefore make explicit when the induced train and test problems share one expert target.

Assumption 5 (Policy-sufficient representation)**.** *There exists a conditional action law $q_\phi^\star(\cdot \mid \xi)$ such that, for both $s \in \{\mathrm{tr}, \mathrm{te}\}$,*

$$q^\star(\cdot \mid X) = q_\phi^\star(\cdot \mid \phi(X)) \qquad P_s^X\text{-almost surely}. \tag{39}$$

Under Assumption 5, define the expert action distribution in anchor–residual coordinates by

$$q_{\mathrm{pair}}^\star(\mathrm{d}\mathbf{a} \mid \boldsymbol{\delta}, \boldsymbol{z}^\dagger) := q_\phi^\star(\mathrm{d}\mathbf{a} \mid \boldsymbol{\delta} + \boldsymbol{z}^\dagger). \tag{40}$$

For reference, an arbitrary encoded conditional $q_\phi^\star$ incurs the information loss induced by representation

$$\varepsilon_{\phi,s}^2 := \mathbb{E}_{X\sim P_s^X} W_2^2\big(q^\star(\cdot \mid X), q_\phi^\star(\cdot \mid \phi(X))\big)\,. \tag{41}$$

Exact policy sufficiency corresponds to $\varepsilon_{\phi,\mathrm{tr}} = \varepsilon_{\phi,\mathrm{te}} = 0$. We use exact sufficiency in the main theorem; Eq. (41) only records the action-distribution information discarded by an approximate representation.

C.2.2 Conditional flow matching in anchor–residual coordinates

The objective at a fixed pair is to sample

$$\boldsymbol{A} \sim q^{\star}_{\mathrm{pair}}(\cdot \mid \boldsymbol{\delta}, \boldsymbol{z}^{\dagger}). \tag{42}$$

Let $p_{\tau|1}(\mathrm{d}y \mid \mathbf{a})$ be the conditional probability path used by flow matching (Lipman et al., 2024), and let $u_{\tau|1}(y \mid \mathbf{a})$ be its conditional vector field. The corresponding marginal path is

$$p_{\tau}(\mathrm{d}y \mid \boldsymbol{\delta}, \boldsymbol{z}^{\dagger}) := \int p_{\tau|1}(\mathrm{d}y \mid \mathbf{a}) q^{\star}_{\mathrm{pair}}(\mathrm{d}\mathbf{a} \mid \boldsymbol{\delta}, \boldsymbol{z}^{\dagger}). \tag{43}$$

with the corresponding marginal vector field:

$$u^{\star}_{\tau}(y \mid \boldsymbol{\delta}, \boldsymbol{z}^{\dagger}) := \frac{\int u_{\tau|1}(y \mid \mathbf{a}) p_{\tau|1}(y \mid \mathbf{a}) q^{\star}_{\mathrm{pair}}(\mathrm{d}\mathbf{a} \mid \boldsymbol{\delta}, \boldsymbol{z}^{\dagger})}{p_{\tau}(y \mid \boldsymbol{\delta}, \boldsymbol{z}^{\dagger})}. \tag{44}$$

Under the standard flow-matching regularity conditions, this vector field generates the marginal path in Eq. (43) and therefore transports the base distribution p_0 to the $q^{\star}_{\mathrm{pair}}(\cdot \mid \boldsymbol{\delta}, \boldsymbol{z}^{\dagger})$ at $\tau = 1$. Appx. C.3 states the precise well-posedness condition used by our guarantee.

To identify the target optimized by CFM, sample

$$\begin{aligned} (\Delta, Z^{\dagger}, \boldsymbol{A}) \sim \overline{P}_s^{\phi,\rho_s}, \qquad \tau \sim \mathcal{U}[0,1], \\ \boldsymbol{A}_{\tau} \sim p_{\tau|1}(\cdot \mid \boldsymbol{A}), \qquad C := (\boldsymbol{A}_{\tau}, \tau), \\ U := u_{\tau|1}(\boldsymbol{A}_{\tau} \mid \boldsymbol{A}). \end{aligned} \tag{45}$$

The random variable U is the conditional vector field evaluated at the sampled path state. The CFM objective regresses the learned vector field against this quantity.

Proposition 6 (CFM regression identity)**.** *For every square-integrable vector field $v(C, \Delta, Z^{\dagger})$,*

$$\begin{aligned} &\mathbb{E}_s\Big[\big\|v(C,\Delta,Z^{\dagger}) - U\big\|_2^2\Big] \\ &\quad = \mathbb{E}_s\Big[\big\|v(C,\Delta,Z^{\dagger}) - u^{\star}(C \mid \Delta, Z^{\dagger})\big\|_2^2\Big] + \mathcal{V}_s, \end{aligned} \tag{46}$$

where

$$u^{\star}(C \mid \Delta, Z^{\dagger}) := \mathbb{E}_s[U \mid C, \Delta, Z^{\dagger}] \tag{47}$$

and

$$\mathcal{V}_s := \mathbb{E}_s\Big[\big\|U - u^{\star}(C \mid \Delta, Z^{\dagger})\big\|_2^2\Big] \tag{48}$$

is independent of v. Under Assumption 5, the conditional expectation in Eq. (47) *equals the marginal vector field in Eq.* (44) *evaluated at $C = (y, \tau)$.*

Proof. Add and subtract $u^{\star}(C \mid \Delta, Z^{\dagger})$ inside the squared norm. The cross term vanishes after conditioning on $(C, \Delta, Z^{\dagger})$ because $u^{\star}$ is the conditional expectation of U. The remaining two terms are the squared error to the conditional expectation and the irreducible conditional variance. The final statement follows from Eq. (44). □

C.2.3 Fixed-context combinatorial extrapolation

For compactness, call

$$c = (y, \tau) \tag{49}$$

a flow context. Fix a context c and output coordinate $k \in [d_A]$. The marginal vector field defines the scalar target

$$h^{\star}_{k,c}(\boldsymbol{\delta}, \boldsymbol{z}^{\dagger}) := u^{\star}_k(c \mid \boldsymbol{\delta}, \boldsymbol{z}^{\dagger}). \tag{50}$$

BFP parameterizes the corresponding predictor as

$$\begin{aligned} h_{\theta,k,c}(\boldsymbol{\delta}, \boldsymbol{z}^{\dagger}) &:= v_{\theta,k}(c \mid \boldsymbol{\delta}, \boldsymbol{z}^{\dagger}) \\ &:= \big\langle f_{\theta,k}(\boldsymbol{\delta}; c), g_{\theta,k}(\boldsymbol{z}^{\dagger}; c) \big\rangle_r . \end{aligned} \tag{51}$$

Let ν_s denote the law of C in regime s, and let

$$\overline{D}^{\mathrm{BFP}}_{s,c} := \mathrm{Law}_s\big(\Delta, Z^\dagger \mid C = c\big) \tag{52}$$

be a regular conditional anchor–residual law. For every fixed (c, k), Eqs. (50) to (52) instantiate the combinatorial-extrapolation problem of Appx. C.1.1: the train and test covariate laws are $\overline{D}^{\mathrm{BFP}}_{\mathrm{tr},c}$ and $\overline{D}^{\mathrm{BFP}}_{\mathrm{te},c}$, the shared regression function is $h^\star_{k,c}$, and the rank-r predictor is $h_{\theta,k,c}$.

Whenever these conditional laws admit the block decomposition of Assumption 2, the matrix-completion identity in Eq. (31) applies directly to their coordinate-wise squared error. Thus, after fixing (c, k), BFP velocity prediction is the same structured low-rank matrix-completion problem reviewed in Appx. C.1.3. Appx. C.3 next states the conditions under which the prior combinatorial-extrapolation bound applies to this family of fixed flow-context problems and converts the resulting vector-field control into an action-distribution guarantee.

C.3 Distributional Extrapolation of BFP

We now complete the argument. We first identify the flow risk controlled by CFM, then combine per-context bilinear transfer with ODE stability to prove the distributional extrapolation theorem. We next state the multi-anchor corollary and conclude by discussing sufficient conditions for the theorem assumptions.

C.3.1 CFM loss induced by BFP

For $s \in \{\mathrm{tr}, \mathrm{te}\}$, define the FM error to the marginal vector field by

$$\mathcal{E}^{\mathrm{FM}}_s(\theta) := \mathbb{E}_{\substack{(\Delta,Z^\dagger)\sim\overline{D}^{\phi,\rho_s}_s,\\ \tau\sim\mathcal{U}[0,1],\\ \boldsymbol{A}_\tau\sim p_\tau(\cdot|\Delta,Z^\dagger)}}\left[\left\|v_\theta(\boldsymbol{A}_\tau,\tau\mid\Delta,Z^\dagger) - u^\star_\tau(\boldsymbol{A}_\tau\mid\Delta,Z^\dagger)\right\|_2^2\right]. \tag{53}$$

By Proposition 6, the expected CFM objective in regime s decomposes as

$$\mathcal{L}^{\mathrm{CFM}}_s(\theta) = \mathcal{E}^{\mathrm{FM}}_s(\theta) + \mathcal{V}_s, \tag{54}$$

where $\mathcal{V}_s$ is independent of θ. The theorem therefore controls the FM error to the marginal vector field; the practical CFM objective differs from it only by the irreducible conditional-variance term. Recall from Eq. (52) that $\overline{D}^{\mathrm{BFP}}_{s,c}$ is the conditional law of the anchor–residual pair given flow context $C = c$, and let ν_s denote the marginal law of C. For output coordinate k, define the fixed-context error

$$\begin{aligned}\mathcal{E}_{s,k}(c;\theta) &:= \mathbb{E}_{(\Delta,Z^\dagger)\sim\overline{D}^{\mathrm{BFP}}_{s,c}}\left[\big(v_{\theta,k}(c\mid\Delta,Z^\dagger) - u^\star_k(c\mid\Delta,Z^\dagger)\big)^2\right],\\ \mathcal{E}_s(c;\theta) &:= \sum_{k=1}^{d_A}\mathcal{E}_{s,k}(c;\theta).\end{aligned} \tag{55}$$

Disintegration of the joint pair–context law gives

$$\mathcal{E}^{\mathrm{FM}}_s(\theta) = \int \mathcal{E}_s(c;\theta)\,\nu_s(\mathrm{d}c). \tag{56}$$

C.3.2 Main theorem: distributional extrapolation

The proof has two steps. First, the prior combinatorial-extrapolation bound transfers the marginal-vector-field error from training to evaluation at each fixed flow context. Second, ODE stability converts evaluation FM error into Wasserstein error of the generated action distribution.

Combinatorial extrapolation of the marginal vector field. Define the bilinear extrapolation factor

$$\Gamma_{\mathrm{bil}} := \kappa^2\left(1 + \frac{64M^4}{\sigma^4}\right). \tag{57}$$

Proposition 7 (Marginal-vector-field extrapolation)**.** *Assume that, for ν_{tr}-almost every flow context c and every output coordinate k, the train and test problems $\overline{D}^{\mathrm{BFP}}_{\mathrm{tr},c}$ and $\overline{D}^{\mathrm{BFP}}_{\mathrm{te},c}$ satisfy* ***(C) Residual Coverage*** *and* ***(S) Bilinear Structure*** *from Appendix C.1.2 with common constants (r, M, σ, κ). Assume also that the fixed-context training errors satisfy*

$$\mathcal{E}_{\mathrm{tr},k}(c;\theta) \le \frac{\sigma^2}{4\kappa} \tag{58}$$

for every relevant (c, k), and that the evaluation flow-context law is covered by the training law,

$$\nu_{\mathrm{te}} \preceq_{\kappa_C} \nu_{\mathrm{tr}}. \tag{59}$$

Then

$$\mathcal{E}^{\mathrm{FM}}_{\mathrm{te}}(\theta) \le \kappa_C \Gamma_{\mathrm{bil}} \mathcal{E}^{\mathrm{FM}}_{\mathrm{tr}}(\theta). \tag{60}$$

Proof. Fix c and k. Appendix C.2.3 identifies $u^\star_k(c \mid \boldsymbol{\delta}, \boldsymbol{z}^\dagger)$ as the shared scalar regression function and $v_{\theta,k}(c \mid \boldsymbol{\delta}, \boldsymbol{z}^\dagger)$ as the rank-r bilinear predictor. Theorem 4 and Eq. (58) therefore give

$$\mathcal{E}_{\mathrm{te},k}(c;\theta) \le \Gamma_{\mathrm{bil}} \mathcal{E}_{\mathrm{tr},k}(c;\theta). \tag{61}$$

Summing over coordinates, integrating with respect to ν_{te}, applying Eq. (59), and using Eq. (56) prove Eq. (60). □

Eq. (58) is context-wise because the prior theorem is applied separately at each fixed context. A small average value of $\mathcal{E}^{\mathrm{FM}}_{\mathrm{tr}}$ alone does not exclude a large error on a rare training context that receives substantial evaluation mass.

From FM error to action-distribution error. Fix an anchor–residual pair $p = (\boldsymbol{\delta}, \boldsymbol{z}^\dagger)$. Let $\mathbf{Y}^\star_\tau$ solve the expert ODE driven by the marginal vector field,

$$\frac{\mathrm{d}\mathbf{Y}^\star_\tau}{\mathrm{d}\tau} = u^\star_\tau(\mathbf{Y}^\star_\tau \mid p), \qquad \mathbf{Y}^\star_0 \sim p_0, \tag{62}$$

and let $\widehat{\mathbf{Y}}_\tau$ solve the learned ODE from the same initial sample,

$$\frac{\mathrm{d}\widehat{\mathbf{Y}}_\tau}{\mathrm{d}\tau} = v_\theta(\widehat{\mathbf{Y}}_\tau, \tau \mid p), \qquad \widehat{\mathbf{Y}}_0 = \mathbf{Y}^\star_0. \tag{63}$$

Proposition 8 (ODE stability)**.** *Assume that the expert ODE is well posed and realizes the marginal path in Eq.* (43)*,*

$$\mathrm{Law}(\mathbf{Y}^\star_\tau) = p_\tau(\cdot \mid p) \qquad \textit{for every } \tau \in [0,1]. \tag{64}$$

Assume also that $v_\theta(\cdot, \tau \mid p)$ is L-Lipschitz in its flow-state argument, uniformly over τ and p. Then

$$\begin{aligned} &W_2^2\Big(\mathrm{Law}(\widehat{\mathbf{Y}}_1), q^\star_{\mathrm{pair}}(\cdot \mid p)\Big) \\ &\quad \le C_{\mathrm{ODE}} \int_0^1 \mathbb{E}_{\mathbf{Y}^\star_\tau \sim p_\tau(\cdot|p)} \Big[\|v_\theta(\mathbf{Y}^\star_\tau, \tau \mid p) - u^\star_\tau(\mathbf{Y}^\star_\tau \mid p)\|_2^2 \Big] \,\mathrm{d}\tau, \\ &\qquad\qquad C_{\mathrm{ODE}} := \exp(2L+1). \end{aligned} \tag{65}$$

Proof. Set $e_\tau := \widehat{\mathbf{Y}}_\tau - \mathbf{Y}^\star_\tau$. Adding and subtracting $v_\theta(\mathbf{Y}^\star_\tau, \tau \mid p)$, using Lipschitzness, and applying Young's inequality yield

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\tau}\|e_\tau\|_2^2 &\le (2L+1)\|e_\tau\|_2^2 \\ &\quad + \|v_\theta(\mathbf{Y}^\star_\tau, \tau \mid p) - u^\star_\tau(\mathbf{Y}^\star_\tau \mid p)\|_2^2 . \end{aligned} \tag{66}$$

Gronwall's inequality and $e_0 = 0$ bound the expected terminal coupling cost by the right-hand side of Eq. (65). Eq. (64) gives $\mathrm{Law}(\mathbf{Y}^\star_1) = q^\star_{\mathrm{pair}}(\cdot \mid p)$, so the squared Wasserstein distance is no larger than this coupling cost. □

For a query x and anchor $x^\dagger$, let $\pi_{\theta,x^\dagger}(\cdot \mid x)$ denote the terminal law of Eq. (63) using $p = (\phi(x) - \phi(x^\dagger), \phi(x^\dagger))$. The test-time BFP policy samples one anchor before integration:

$$\pi_{\theta,\rho_{\mathrm{te}}}(\cdot \mid x) := \int \pi_{\theta,x^\dagger}(\cdot \mid x)\, \rho_{\mathrm{te}}(\mathrm{d}x^\dagger \mid x, \mathcal{D}). \tag{67}$$

Theorem 9 (Distributional extrapolation of BFP)**.** *Suppose Assumption 5 holds, the assumptions of Proposition 7 hold, and the ODE assumptions of Proposition 8 hold. Then*

$$\mathcal{R}^{\mathrm{dist}}_{\mathrm{te}}(\pi_{\theta,\rho_{\mathrm{te}}}) \le C_{\mathrm{ODE}} \kappa_C \Gamma_{\mathrm{bil}} \mathcal{E}^{\mathrm{FM}}_{\mathrm{tr}}(\theta). \tag{68}$$

Proof. Under Assumption 5, every anchor–residual representation of a query x has the same target $q^\star_{\mathrm{pair}}(\cdot \mid p) = q^\star(\cdot \mid x)$. Apply Proposition 8 for each test query and sampled anchor, then average the resulting coupling over $X^\dagger \sim \rho_{\mathrm{te}}(\cdot \mid X, \mathcal{D})$ and $X \sim P^X_{\mathrm{te}}$. This gives

$$\mathcal{R}^{\mathrm{dist}}_{\mathrm{te}}(\pi_{\theta,\rho_{\mathrm{te}}}) \le C_{\mathrm{ODE}} \mathcal{E}^{\mathrm{FM}}_{\mathrm{te}}(\theta). \tag{69}$$

Proposition 7 then yields Eq. (68). □

C.3.3 Corollary: multi-anchor test-time policies

Theorem 9 already permits a stochastic test-time anchor policy. The following corollary makes the resulting policy distribution explicit.

Corollary 10 (Mixture over anchors)**.** *For a fixed query x, let $\rho(\mathrm{d}a \mid x)$ be any anchor policy and let $\pi_{\theta,a}(\cdot \mid x)$ be the terminal distribution obtained by fixing anchor a throughout the ODE solve. Then*

$$\pi_{\theta,\rho}(\cdot \mid x) = \int \pi_{\theta,a}(\cdot \mid x)\, \rho(\mathrm{d}a \mid x), \tag{70}$$

and, for every target distribution $q(\cdot \mid x)$,

$$\begin{aligned} &W_2^2(\pi_{\theta,\rho}(\cdot \mid x), q(\cdot \mid x)) \\ &\quad \le \int W_2^2(\pi_{\theta,a}(\cdot \mid x), q(\cdot \mid x))\ \rho(\mathrm{d}a \mid x). \end{aligned} \tag{71}$$

Proof. Sample the anchor from ρ and then sample from the corresponding anchor-conditioned terminal law, which gives Eq. (70). For each anchor, choose an optimal coupling between $\pi_{\theta,a}(\cdot \mid x)$ and $q(\cdot \mid x)$. Sampling the anchor first and then sampling from its coupling produces a valid coupling for the mixture whose expected transport cost is the right-hand side of Eq. (71). □

The anchor must be sampled before ODE integration and held fixed throughout the solve. Averaging the anchor-conditioned vector fields instead defines a different ODE and does not generally produce the mixture in Eq. (70).

C.3.4 Affine-Gaussian inheritance of bilinear structure

The main theorem requires policy sufficiency, context-wise bilinear structure and combinatorial coverage, flow-context coverage, and ODE regularity. These conditions concern different parts of BFP and should not be conflated.

The following proposition provides one sufficient setting for the bilinear-realizability part of **(S) Bilinear Structure**. It is an explanatory result and is not required by the implementation.

Proposition 11 (Affine-Gaussian inheritance of bilinearity)**.** *Let $p = (\boldsymbol{\delta}, \boldsymbol{z}^\dagger)$ and suppose the pair-conditioned expert endpoint is*

$$\boldsymbol{A} \mid p \sim \mathcal{N}(m(p), \Sigma_1), \tag{72}$$

where Σ_1 does not depend on p. Assume an affine Gaussian probability path whose conditional vector field has the form

$$u_{\tau|1}(y \mid \boldsymbol{A}) = c_\tau(y) + B_\tau \boldsymbol{A}. \tag{73}$$

Then there exist matrices K_τ and L_τ, independent of p, such that

$$\mathbb{E}[\boldsymbol{A} \mid \boldsymbol{A}_\tau = y, p] = K_\tau y + L_\tau m(p), \tag{74}$$

and therefore

$$u_\tau^\star(y \mid p) = \widetilde{c}_\tau(y) + \widetilde{B}_\tau m(p). \tag{75}$$

If every endpoint-mean coordinate has a bilinear realization

$$m_\ell(p) = \left\langle f_\ell^\star(\boldsymbol{\delta}), g_\ell^\star(\boldsymbol{z}^\dagger) \right\rangle_{r_\ell}, \tag{76}$$

then output coordinate k of the marginal vector field has bilinear rank at most

$$1 + \sum_{\ell : [\widetilde{B}_\tau]_{k\ell} \neq 0} r_\ell. \tag{77}$$

If $\widetilde{B}_\tau$ is diagonal, the rank is at most $r_k + 1$.

Proof. Eq. (72) and the affine Gaussian path form a linear-Gaussian model, so the posterior mean of the endpoint is affine in the observed path state and in the prior mean, yielding Eq. (74). Taking the conditional expectation of Eq. (73) gives Eq. (75). For coordinate k, expand $[\widetilde{B}_\tau m(p)]_k$ over the endpoint coordinates, concatenate the scaled bilinear features for every nonzero term, and add one constant feature for $\widetilde{c}_{\tau,k}(y)$. The resulting inner product has the rank in Eq. (77). □

C.3.5 When do the remaining assumptions hold?

Representation sufficiency. Assumption 5 holds when $\phi(X)$ is a sufficient statistic for expert behavior: raw conditions mapped to the same encoded query must have the same expert action distribution. The representation information loss in Eq. (41) measures violations at the action distribution.

Combinatorial coverage and feature excitation. These are properties of the anchor–residual problems constructed by $(\phi, \rho_{\mathrm{tr}}, \rho_{\mathrm{te}})$. The train-time policy must expose the three product blocks required by Assumption 2 and sufficiently excite every retained residual-side and anchor-side feature direction in Assumption 3. The test-time policy must represent an evaluation query using factors covered by those training blocks. The learned shortlist, non-collapse regularization, and compact representation are mechanisms intended to make these properties plausible; they do not certify them.

Flow-context coverage. A simple sufficient condition follows when training and evaluation use the same endpoint-conditioned probability-path kernel. Let

$$\overline{q}_s(\mathrm{d}\mathbf{a}) := \int q^\star(\mathrm{d}\mathbf{a} \mid x) P_s^X(\mathrm{d}x) \tag{78}$$

be the aggregate expert endpoint law. If

$$\overline{q}_{\mathrm{te}} \preceq_{\kappa_A} \overline{q}_{\mathrm{tr}}, \tag{79}$$

then the common path kernel preserves this domination and implies

$$\nu_{\mathrm{te}} \preceq_{\kappa_A} \nu_{\mathrm{tr}}. \tag{80}$$

Indeed, for every measurable context set B, integrating the same conditional probability $\Pr(C \in B \mid \boldsymbol{A} = \mathbf{a})$ against Eq. (79) gives Eq. (80). This is sufficient for Eq. (59), but it is not implied merely by bounded actions or by overlapping Gaussian support.

ODE regularity. Proposition 8 uses standard ODE conditions: the expert marginal vector field must realize the prescribed path, and the learned vector field must be uniformly Lipschitz in the flow state over the relevant domain. These assumptions concern stable flow integration and are separate from the combinatorial-extrapolation assumptions on the anchor–residual coordinates.

D RETRIEVAL POLICY METHOD DETAILS

Algorithm 1 Training

Require: $\mathcal{D}$; encoder ϕ; predictor v_θ; scorer s_ψ
repeat
  sample train queries $(\boldsymbol{x}, \boldsymbol{A})$ and build $\mathcal{C}(\boldsymbol{x})$
  compute p_ψ by equation 11
  update θ with equation 12
  update EMA predictor $\bar{\theta}$
  if retrieval-update step **then**
    sample held-out queries and rebuild $\mathcal{C}(\boldsymbol{x})$
    evaluate candidate losses with $\bar{\theta}$
    update ψ with equation 13
  end if
until convergence

Algorithm 2 Inference

Require: query $\boldsymbol{x}$; $\mathcal{D}$; ϕ, v_θ, s_ψ
retrieve demonstration $j^\star$ using equation 14
build the phase-compatible $\mathcal{C}(\boldsymbol{x})$ within j^*
compute anchor weights $\omega_m(\boldsymbol{x})$ from the scorer
sample $M \sim \mathrm{Cat}(\omega(\boldsymbol{x}))$
form $\boldsymbol{z}_M^\dagger$ and $\boldsymbol{\delta}_M$
sample $\boldsymbol{A}_0 \sim p_0$
integrate $\dot{\boldsymbol{A}}_\tau = v_\theta(\boldsymbol{A}_\tau, \tau \mid \boldsymbol{\delta}_M, \boldsymbol{z}_M^\dagger)$
set $\boldsymbol{A} = \boldsymbol{A}_1$ and execute its first T_a actions
re-observe and repeat

This section specifies the retrieval policy used by BFP, that includes the specific phase-conditioned candidate set algorithm, scorer architectures, and predictor–scorer updates.

Phase-conditioned Candidate Set $\mathcal{C}(\boldsymbol{x})$. For query (x_t, A_t) from trajectory i at time t, let f_t denote its phase, $\mathrm{seg}_j(f) = [s_j(f), e_j(f)]$ the phase-f segment of demonstration j, and $\mathrm{el}_t = t - s_i(f_t)$ the elapsed time within the current phase. We first select J anchor demonstrations that reach phase f_t, uniformly sampled phase-compatible demonstrations, excluding trajectory i.

For each selected demonstration j, we align by elapsed phase time and define

$$\rho_j = s_j(f_t) + \min\{\mathrm{el}_t, \ell_j(f_t) - 1\}, \qquad W_j = [\rho_j - h_j, \rho_j + h_j] \cap \mathrm{seg}_j(f_t), \qquad h_j = \lfloor w(\ell_j(f_t) - 1) \rfloor . \tag{81}$$

We sample M frames from each W_j, yielding $\mathcal{C}(x_t) = \{x_{tm}^\dagger\}_{m=1}^K$ with $K = JM$.

Scorer Architecture s_ψ. For candidate $\boldsymbol{x}_{tm}^\dagger \in \mathcal{C}(\boldsymbol{x}_t)$, let

$$\boldsymbol{z}_{tm}^\dagger = \phi(\boldsymbol{x}_{tm}^\dagger), \qquad \boldsymbol{\delta}_{tm} = \phi(\boldsymbol{x}_t) - \boldsymbol{z}_{tm}^\dagger.$$

MLP scores each candidate from features $[\boldsymbol{\delta}_{tm}/\sigma,\ \boldsymbol{\delta}_{tm}/\|\boldsymbol{\delta}_{tm}\|_2,\ \log \|\boldsymbol{\delta}_{tm}\|_2,\ \boldsymbol{z}_{tm}^\dagger]$. The resulting responsibility is

$$q_{tm} = p_\psi(m \mid x_t, \mathcal{C}(x_t)) = \mathrm{softmax}_m \left(\frac{s_\psi(\boldsymbol{\delta}_{tm}, \boldsymbol{z}_{tm}^\dagger)}{\tau_s} \right), \tag{82}$$

where τ_s is learned and initialized to 1.

Predictor update. We sample predictor queries from $\mathcal{B} \cup \mathcal{C}$, construct $\mathcal{C}(\boldsymbol{x}_i)$, and compute the responsibilities in Eq. (82). With these responsibilities treated as fixed, the predictor minimizes

$$\mathcal{L}_\theta = \frac{1}{|\mathcal{B}_\theta|} \sum_{i \in \mathcal{B}_\theta} \sum_{m=1}^{K} \mathrm{sg}[w_{im}]\, \ell_{\mathrm{BFP}}(\theta; \boldsymbol{x}_i, \boldsymbol{x}_{im}^\dagger, \boldsymbol{A}_i), \tag{83}$$

where w_{im} denotes the responsibility after applying the scheduled predictor-weight floor.

Scorer update. Every $K_\theta = 5$ predictor updates, we draw a fresh batch $\mathcal{B}_\psi$ of 32 queries from the scorer pool $\mathcal{A}$. For each query and candidate, we evaluate the BFP loss using an exponential-moving-average predictor $\bar{\theta}$. We average over $R_\psi = 4$ flow contexts,

$$\bar{\ell}_{im} = \frac{1}{R_\psi} \sum_{r=1}^{R_\psi} \ell_{\mathrm{BFP}}^{(r)}(\bar{\theta}; \boldsymbol{x}_i, \boldsymbol{x}_{im}^\dagger, \boldsymbol{A}_i), \tag{84}$$

using the same sampled flow contexts across candidates of a query. This shared sampling makes candidate losses directly comparable.

Table 7: **Key hyperparameters for simulation experiments.** Anchor parameters apply only to BFP.

Hyperparameter	Value
Optimizer	Adam, $\beta = (0.9, 0.999)$
Learning rate (BFP predictor/scorer)	10^{-4}
Batch size	32 queries
Gradient clipping	Global norm 1.0
EMA power / maximum decay	$0.75/0.9999$
Training steps	120k (Relocate-D0), 420k (Relocate-D1) 300k (remaining tasks)
ODE solver / integration steps	Euler / 10
Action horizon / execution horizon	MuJoCo: $1/1$; ManiSkill: $16/8$
Bilinear rank	32; ablations: 128, 512
Anchor demonstrations J	$\{1, 4\}$
Frames per demonstration M	16
Phase window w	$\{0.25, 0.5\}$
Scorer update interval	5 predictor steps
Maximum predictor weight floor	0.5
Training seeds	3
Evaluation conditions per seed	100 ID + 100 OOD

Because the absolute CFM-loss scale varies across queries, candidate losses are robustly normalized within each shortlist:

$$\widetilde{\ell}_{im} = \frac{\bar{\ell}_{im} - \mathrm{med}_n(\bar{\ell}_{in})}{\mathrm{MAD}_n(\bar{\ell}_{in}) + \epsilon}. \tag{85}$$

The scorer minimizes

$$\mathcal{L}_\psi = \frac{1}{|\mathcal{B}_\psi|} \sum_{i \in \mathcal{B}_\psi} \sum_{m=1}^{K} q_{im}\, \mathrm{sg}[\widetilde{\ell}_{im}] + \lambda_{\mathrm{nc}} \mathcal{R}_{\mathrm{nc}}, \tag{86}$$

so candidates with lower held-out BFP loss receive higher probability.

To prevent collapse onto a small number of candidates, we use an entropy hinge

$$\mathcal{R}_{\mathrm{nc}} = \frac{1}{|\mathcal{B}_\psi|} \sum_{i \in \mathcal{B}_\psi} [\eta - H(q_i)]_+, \qquad \eta = 0.6 \log K, \qquad \lambda_{\mathrm{nc}} = 1. \tag{87}$$

Predictor and scorer parameters are optimized separately, and gradients are not propagated through the responsibilities in Eq. (83) or through candidate losses in Eq. (86).

E Numerical Example: Distributional Extrapolation

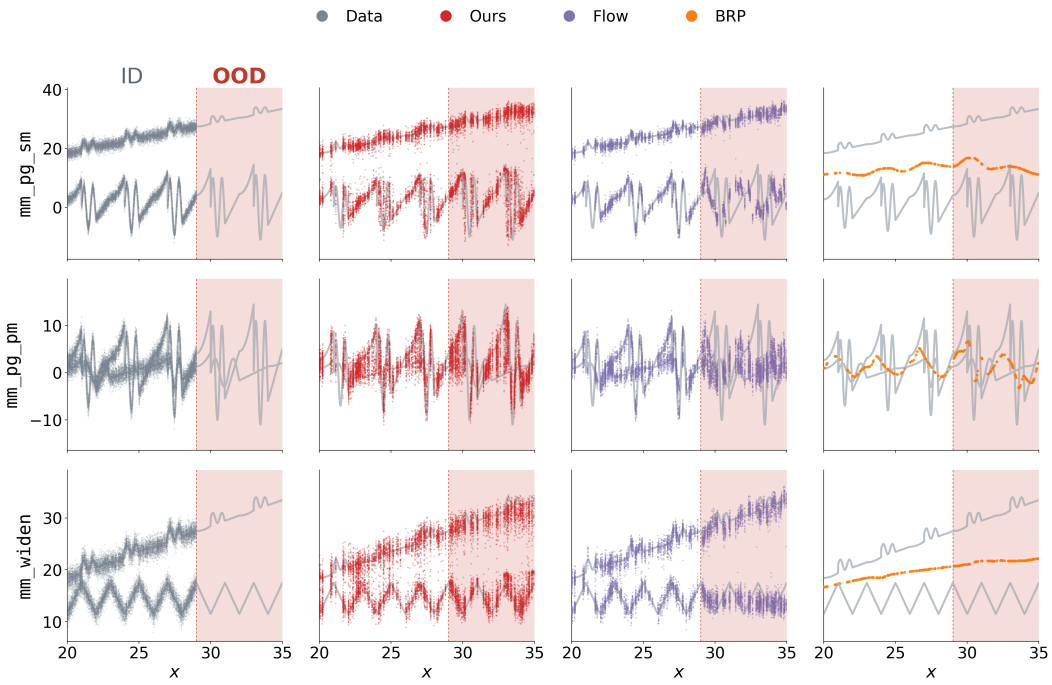


Figure 17: **Distributional extrapolation sample visualization.** Rows correspond to different tasks: `mm_pg_sm`, `mm_pg_pm`, and `mm_widen`, while columns show the samples of training data (gray), BFP (red), FLOW (purple), and BRP (orange).

This section specifies the controlled experiment summarized in Fig. 2 and Tab. 8. The figure in the main text reports `mm_pg_sm`, while we report two additional problems. The experiment separates deterministic bilinear extrapolation, conditional generation without a relational extrapolation rule, and their combination in BFP.

E.1 Expert Conditional Distribution

We define the expert conditional distribution $q^\star(\cdot \mid x)$ using pairs of analytical mode functions. For $T > 0$, let

$$r_T(x) := x - T \left\lfloor \frac{x}{T} \right\rfloor \in [0, T) \tag{88}$$

denote the remainder of x modulo T. The shared base profile $g : [0, 3) \to \mathbb{R}$ is

$$g(u) := \begin{cases} \sin(10u), & 0 \le u < 1, \\ \sin(10) + (u - 1), & 1 \le u < 2, \\ \sin(10) + 1 + (u - 2)^2, & 2 \le u < 3. \end{cases} \tag{89}$$

We additionally define the time warp

$$w(x) := \begin{cases} r_9(x), & r_9(x) < 3, \\ \big(r_9(x) - 3\big)/2, & r_9(x) \ge 3, \end{cases} \tag{90}$$

and the triangular wave

$$\mathrm{tri}(x) := \begin{cases} r_1(x), & \lfloor x \rfloor \text{ even}, \\ 1 - r_1(x), & \lfloor x \rfloor \text{ odd}. \end{cases} \tag{91}$$

Method	$\mathrm{NW}_2 \downarrow$ ID	$\mathrm{NW}_2 \downarrow$ OOD	Balance ↓ ID	Balance ↓ OOD
mm_pg_sm				
Oracle	0.10±0.01	0.11±0.01	0.02±0.00	0.02±0.00
Flow	0.23±0.01	0.30±0.01	0.09±0.00	0.07±0.01
BRP	0.51±0.00	0.53±0.00	0.50±0.00	0.50±0.00
BFP	**0.15±0.01**	**0.17±0.00**	**0.03±0.00**	**0.03±0.01**
mm_pg_pm				
Oracle	0.07±0.00	0.06±0.00	0.02±0.00	0.02±0.00
Flow	**0.25±0.00**	0.97±0.01	0.09±0.00	0.33±0.00
BRP	1.13±0.05	1.24±0.08	0.50±0.00	0.50±0.00
BFP	0.43±0.01	**0.58±0.03**	**0.07±0.01**	**0.20±0.00**
mm_widen				
Oracle	0.07±0.00	0.09±0.01	0.02±0.00	0.02±0.00
Flow	0.19±0.00	0.25±0.00	0.09±0.00	**0.08±0.00**
BRP	0.53±0.00	0.51±0.00	0.50±0.00	0.50±0.00
BFP	**0.10±0.01**	**0.21±0.01**	**0.03±0.00**	**0.08±0.00**

Table 8: **Controlled distributional extrapolation.** Mean ± standard deviation over three seeds. Bold denotes the best learned method; Oracle is the finite-sampling floor.

Problem	f_1	f_2
mm_pg_sm	f_{pg}	f_{sm}
mm_pg_pm	f_{pg}	f_{pm}
mm_widen	f_{sm}	f_{sbl}

Table 9: **Distributional extrapolation problem setting.** Each problem combines two mode functions from equation 92.

The four mode functions are

$$f_{\mathrm{pg}}(x) := \left\lfloor \frac{x}{3} \right\rfloor g(r_3(x)), \qquad f_{\mathrm{sm}}(x) := 3\left\lfloor \frac{x}{3} \right\rfloor + g(r_3(x)),$$
$$f_{\mathrm{pm}}(x) := \left\lfloor \frac{x}{9} \right\rfloor g(w(x)), \qquad f_{\mathrm{sbl}}(x) := 6\,\mathrm{tri}(x) + 11.5. \tag{92}$$

For each problem, the expert conditional distribution is

$$q^\star(y \mid x) = \frac{1}{2}\mathcal{N}\big(f_1(x), \sigma^2\big) + \frac{1}{2}\mathcal{N}\big(f_2(x), \sigma^2\big), \qquad \sigma = 1, \tag{93}$$

where f_1 and f_2 are selected according to Tab. 9.

The training and OOD condition distributions are

$$P_{\mathrm{tr}}^X = \mathrm{Unif}([20, 29]), \qquad P_{\mathrm{te}}^X = \mathrm{Unif}([29, 35]). \tag{94}$$

Thus, evaluation changes only the support of the condition while preserving the same expert conditional law.

E.2 Experimental Details

Setting. We sample $2,000$ training conditions from P_{tr}^X and 10 independent targets from $q^\star(\cdot \mid x)$ for each condition, yielding $20,000$ training data. Evaluation uses 100 ID conditions from P_{tr}^X and 100 OOD conditions from P_{te}^X. We compare a standard Flow model, BRP, and BFP. All models use depth-3, width-1,000 MLPs and are trained for 300 epochs, 3 random seeds. Flow and BFP use 50 Euler steps for generation. BRP and BFP use $r = 16$ for rank.

Metric. For each query x, we draw $K = 500$ samples and form the empirical policy law $\widehat{q}_{\theta,K}(\cdot \mid x)$. Our primary metric is the Wasserstein distance to the true expert conditional, normalized by the separation between its two modes:

$$\mathrm{NW}_2(x) := \frac{W_2(\widehat{q}_{\theta,K}(\cdot \mid x), q^\star(\cdot \mid x))}{|f_1(x) - f_2(x)|}. \tag{95}$$

We also report mode-balance error. Let

$$f_{\mathrm{low}}(x) := \min\{f_1(x), f_2(x)\}, \qquad f_{\mathrm{high}}(x) := \max\{f_1(x), f_2(x)\},$$

and define the empirical probability assigned to the lower mode as

$$\widehat{p}_{\mathrm{low}}(x) := \frac{1}{K}\sum_{k=1}^{K} \mathbb{1}[|\widehat{y}_k(x) - f_{\mathrm{low}}(x)| \le |\widehat{y}_k(x) - f_{\mathrm{high}}(x)|]\,. \tag{96}$$

The mode-balance error is

$$\mathrm{Bal}(x) := \left|\widehat{p}_{\mathrm{low}}(x) - \frac{1}{2}\right|. \tag{97}$$

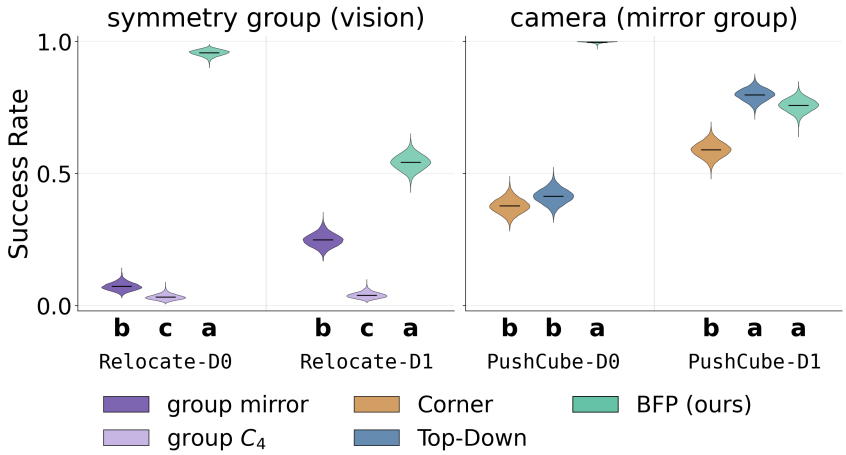
Because both metrics have a finite-sample floor, we measure the *oracle row* by applying the same evaluation to samples drawn directly from $q^\star(\cdot \mid x)$.

Result. BFP achieves the lowest OOD NW_2 on all three problems, with a mean of 0.321 compared with 0.506 for Flow, a 36.6% reduction. It also yields better OOD mode balance on two problems and ties Flow on the third. In contrast, BRP cannot represent both modes and performs worst in distributional error. These results show that bilinear structure provides the extrapolation rule, while the generative flow head is necessary to preserve the conditional distribution. We visualize the full samples at Fig. 17.

F BASELINE INSTANTIATION AND ABLATION RESULTS

	FLOW		CTA		BFP	
Encoder	ID	OOD	ID	OOD	ID	OOD
Scratch	96.9	36.9	85.4	9.2	59.4	**45.0**
R3M-34 (frozen)	96.3	30.2	50.4	6.8	99.4	**76.4**

(a) **Is PVR sufficient for extrapolation?**



(b) **BFP wins hand-specified Extrapolation Rule.**

Figure 18: **Sensitivity to representation and prescribed extrapolation rules.** (a) ID/OOD success rate under visuomotor, averaged over five tasks. Scratch uses a jointly trained ResNet-34 for FLOW and BFP and IMPALA encoder for CTA. Bold denotes the best-performing method within each encoder setting and split. (b) OOD sensitivity of EQUIDP to the prescribed symmetry and camera viewpoint.

Overview. Our main result (Fig. 4) uses the baseline instantiations we found most appropriate for unseen-goal generalization. Because several methods were not originally designed for this setting, we validate the implementation choices that affect OOD performance. These studies also show that (i) BFP's gains are not explained by pretrained visual representations alone, and (ii) learning the extrapolation rule from demonstrations can be preferable to prescribing it a priori.

Setup. We match the demonstration data, evaluation episodes, action representation, prediction and execution horizons, and policy architecture whenever compatible with the original method. We retain method-specific components when they define the baseline's generalization mechanism, such as the equivariant structure of EQUIDP or the learned representation of CTA. Tab. 10 summarizes the resulting comparison and the main design axes separating each baseline from BFP.

F.1 IMPLEMENTATION DETAILS

Flow Policy. FLOW uses the same frozen R3M representation, conditional flow-matching objective, action parameterization, and policy backbone as BFP, but directly conditions on the observation and goal without an anchor–residual extrapolation rule.

Bilinear Regression Policy. BRP uses the same frozen R3M representation and policy backbone as BFP, while retaining the synchronized anchor policy of bilinear transduction. It replaces conditional flow matching with deterministic squared-loss action regression.

DARP. We retain DARP's local anchor retrieval and residual-based extrapolation rule, but use conditional flow matching for a matched generative comparison. We use the same frozen R3M representation and policy backbone as BFP and tune its method-specific retrieval hyperparameters.

Table 10: **Design axes of the main baselines.** The table separates three components of each method: the observation representation, the extrapolation rule used beyond demonstrated goals, and the prediction model used to generate actions. This decomposition highlights which components are matched to BFP and which define each baseline's extrapolation mechanism.

Method	Representation	Extrapolation Rule	Prediction Model
BFP	PVR	learned anchor–residual bilinear	conditional flow
FLOW	PVR	none	conditional flow
BRP	PVR	synchronized anchor–residual bilinear	deterministic regression
DARP	PVR	local retrieved residual	conditional flow
EQUIDP	Unfrozen	prescribed geometric equivariance	conditional flow
CTA	Unfrozen	learned temporal-distance analogy	hierarchical policy

CTA. We retain CTA's native hierarchical policy and learned IMPALA representation, which jointly define its temporal-distance analogy mechanism. Its low-level policy predicts primitive actions, so we evaluate CTA with $T_p = 1$ in the main comparison. For Tp=16 task, we use Q-chunking.

Equivariant Policy. EQUIDP uses conditional flow matching while retaining its native equivariant architecture and task-specific geometric group action. We use the common benchmark camera in the main comparison; the effect of symmetry and viewpoint choice is studied below

F.2 ADDITIONAL BASELINE STUDIES

Is pretrained visual representation sufficient? We test whether BFP's visuomotor gains can be explained by pretrained visual representations (PVRs) alone. As shown in Fig. 18a, frozen R3M does not improve FLOW over a jointly trained ResNet-34 (30.2% vs. 36.9% OOD), and degrades CTA relative to its from-scratch IMPALA encoder (6.8% vs. 9.2%). In contrast, R3M substantially improves BFP from 45.0% to 76.4% OOD success. Thus, PVR is not itself sufficient for extrapolation; rather, its benefit in BFP appears to come from providing a representation in which anchor–residual differences form a useful coordinate system for bilinear extrapolation.

How sensitive is a prescribed extrapolation rule? Fig. 18 evaluates two choices required by EQUIDP: the prescribed symmetry group and the camera viewpoint. On `Relocate-D1`, changing the geometric group reduces OOD success from 26% to 6%, while on `PushCube-D1`, using a symmetry-aligned top-down camera improves OOD success from 59% to 80%. These choices are not reliably identified by ID performance. In the same settings, BFP remains competitive or stronger without requiring the deployment transformation to be specified in advance. This sensitivity highlights the value of learning the extrapolation rule from demonstrations rather than prescribing it a priori.

F.3 PRE-ROLLOUT EXTRAPOLATION DIAGNOSIS

We evaluate whether the two theory-motivated diagnostics in Fig. 5a can anticipate extrapolation error without executing the learned policy in closed loop. To do so, we first construct a privileged measure of OOD prediction error, which is used only as the response variable for evaluating the diagnostics.

Privileged OOD error. For each task, we collect 8 expert rollouts under held-out goals. These rollouts provide OOD observation–action pairs: the observations are those induced by unseen goals, while the corresponding actions are labeled by the expert policy. This privileged OOD information is *not* provided to either diagnostic; it is used only to measure whether the diagnostic agrees with the error that would be observed if OOD expert labels were available. For each learned checkpoint, we replay the OOD observations open loop and evaluate its flow-matching (FM) loss against the expert action targets. The velocity-matching MSE is averaged over the fixed flow-time grid $\tau \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ and frozen Gaussian noise samples. Averaging this loss over the evaluated OOD observations gives the checkpoint-level OOD error used in **(S) Bilinear Structure**; for **(C) Residual Coverage**, we retain the corresponding FM error for each individual OOD query.

(S) Bilinear Structure : validation error for checkpoint selection. The diagnostic available without OOD labels is the FM error on a held-out in-distribution validation split. We ask whether this held-out ID error can identify, among policies trained for the same task, which checkpoint will have lower privileged OOD error. To isolate this model-selection question, checkpoints are compared only within same pair of the task and observation space. The analysis contains five tasks under state and visuomotor observations. We report the result across 180 trained checkpoints in total.

Within each task, ID and OOD FM errors are independently converted to percentile ranks. For visualization in Fig. 5a, the within-pool ID-error ranks are pooled and divided into quintiles, and we plot the mean OOD-error percentile rank in each quintile. This evaluation therefore asks whether held-out ID error preserves the *ordering* of checkpoints that would be obtained from privileged OOD error, rather than their absolute values.

(C) Residual Coverage : residual coverage for goal selection. The second diagnostic operates within a fixed BFP checkpoint and asks whether the residual induced for a particular OOD query is supported by the residuals encountered during training. For each checkpoint, we construct a reference bank of $20{,}000$ training residuals. A training query $\boldsymbol{x}$ is paired with an anchor $\boldsymbol{x}^\dagger$ sampled from the learned train-time anchor policy, and we store

$$\boldsymbol{\delta} = \phi(\boldsymbol{x}) - \phi(\boldsymbol{x}^\dagger). \tag{98}$$

The resulting bank therefore represents the residual distribution induced by the learned anchor policy on training data.

For an OOD query, the deployment-time anchor policy selects $\boldsymbol{x}^\dagger$ and forms its query residual $\boldsymbol{\delta}_{\mathrm{OOD}}$ in the same representation space. We score the coverage of this residual against the training residual bank using the realism metric (Kynkäänniemi et al., 2019); larger scores indicate that the query residual lies more deeply inside the local support of the training residual distribution.

Within each checkpoint, OOD queries are ranked by residual coverage and divided into quintiles, while their FM errors are independently converted to within-checkpoint percentile ranks. Fig. 5a aggregates the final rank-32 checkpoint from 42 trained policies—seven simulation settings, two observation spaces, and three seeds. For each coverage quintile, we report the median error rank across these 42 checkpoints.