# Transport Discrepancy as a Reliability Signal for Vision-Language-Action Models

Wanpeng Zhang[1,3], Ye Wang[2,3], Hao Luo[1,3], Haoqi Yuan[1,3], Yicheng Feng[1,3], Chaoyi Xu[1,3], Sipeng Zheng[3], Qin Jin[2], and Zongqing Lu[1,3]✉

[1] Peking University, Beijing, China
[2] Renmin University of China, Beijing, China
[3] BeingBeyond, Beijing, China
https://research.beingbeyond.com/dig

**Abstract.** Vision-language-action (VLA) models that generate continuous action chunks via flow matching lack an internal signal for judging whether a given prediction is reliable. Distribution shift and long-horizon rollouts can push backbone representations away from the region the action head decodes reliably, yet the policy has no mechanism to detect or react to this drift. We observe that the cost of transporting observation features to the action representation in a shared feature space rises precisely when such drift occurs, providing a per-step reliability estimate without extra supervision. Building on this observation, we propose `DiG` (Discrepancy Gate), a lightweight plug-in module for flow-matching VLA policies. `DiG` computes a sliced Wasserstein transport cost between backbone features and the action expert's own input projection, maps it through an exponential gate, and uses the gate to modulate both a residual feature refinement and the training loss. At inference time, the gate enables `DiG-Refine`, an iterative refinement process that corrects action chunks before execution. Experiments on both simulation and real-world scenarios show that `DiG` consistently improves success rates, with the largest gains under distribution shift and on long-horizon tasks.

**Keywords:** Vision-Language-Action · Flow Matching · Transport Discrepancy · Robot Manipulation

## 1 Introduction

Vision-language-action (VLA) models map images and language instructions to robot controls by combining a pretrained vision-language backbone with an action head [5, 8, 23]. Recent systems often generate continuous action chunks with diffusion or flow matching inside a dedicated action expert [4, 5, 21]. In flow matching, the action expert learns a time-conditioned velocity field whose ODE *transports* a simple noise distribution into the action distribution. This transport view is the mechanism behind action generation, and it also provides a natural way to detect when the policy is likely operating out of distribution: the cost of

✉ Correspondence to Zongqing Lu: lu@beingbeyond.com.

transporting observation features to the action representation rises when they become misaligned.

For a fixed backbone and action head, only part of the backbone representation space can be reliably decoded into correct actions. This subset behaves like a low-dimensional *decodable manifold*. Distribution shift and long-horizon rollouts can push the backbone features away from that manifold, so the action head still produces a chunk, but the prediction is poorly conditioned. Errors then compound across steps.

A practical implication is the need for a per-step reliability estimate. Such a signal can serve two roles. During training, it can reduce the influence of demonstration pairs that are explained by shortcuts or spurious cues [18, 55]. At inference time, it can indicate when a predicted chunk is inconsistent with the current observation and should be corrected before execution.

This paper introduces `DiG` (Discrepancy Gate), a plug-in module for flow-matching VLA policies. `DiG` turns the transport view of flow matching into an *internal reliability signal*. It computes a transport discrepancy between (i) the backbone feature distribution of the current observation and (ii) a compact action representation obtained by the same action expert. The discrepancy is instantiated as a sliced 2-Wasserstein transport cost [6,24]. A monotone mapping produces a gate value that modulates a lightweight residual refinement of the backbone features. The same gate also reweights the flow-matching loss. During backpropagation, $\mathrm{sg}(\cdot)$ is applied only to the gate. This blocks direct optimization of the discrepancy while keeping learning signals from the original flow-matching loss. At deployment, `DiG-Refine` runs a short inference-time refinement loop that reuses the same gate with predicted chunks. The module attaches to the context features consumed by the action expert. It is compatible with both shared-transformer designs that use prefix KV caches and two-stage designs with a separate action expert.

The contributions of our paper are as follows: 1) A transport-based reliability signal for flow-matching VLA policies, realized as a lightweight module and a matching inference-time procedure that improve robustness without changing the flow-matching target. 2) A theoretical analysis that connects the exponential gate to a local manifold noise model (Proposition 1) and formalizes how gating suppresses shortcut components under a mixture contamination model (Theorem 1). 3) Extensive experiments on both simulation and real-robot tasks, including perturbation and sensitivity analyses, to verify the effectiveness of `DiG`.

## 2 Related Work

VLA policies build on progress in vision-language models (VLMs). Large multimodal backbones couple language reasoning [3, 49, 50] with strong visual encoders [16, 36, 43, 56] and exhibit broad instruction-following behavior [1, 29, 30, 32,53,57,60,64]. VLA work transfers these representations to robot control. RT-1 and RT-2 scale transformer policies with web data and large robot datasets [7,8]. OpenVLA provides an open 7B-scale model and benchmark suite [22, 23]. Other

lines explore modular and multi-embodiment policies [14,45]. Several recent systems generate actions with diffusion or flow matching to model multimodality [4, 5, 12, 21]. GR-2 and GR-3 leverage video pretraining for robot control [9, 10]. SpatialVLA and VIPA-VLA incorporate spatial representations [17, 42]. CoT-VLA adds visual chain-of-thought reasoning [63]. Otter focuses on fine-grained visual features [20]. OneTwoVLA studies adaptive reasoning in a unified architecture [26]. Being-H0 incorporates physical instruction tuning for better motion understanding [33], and Being-H0.5 scales human-centric learning with a unified action space for cross-embodiment robot control [35]. Several recent systems use predictive representations to support future-aware robot control [34, 37, 44, 59, 61, 62]. These systems still struggle under distribution shift, especially in long-horizon tasks, and typically lack an internal reliability estimate [54].

Flow matching provides the action-generation mechanism for many VLA policies. It learns a time-conditioned velocity field without simulating a forward process [27]. Related work includes conditional flow matching [48], rectified flow [31], improved probability paths [29], and stochastic interpolants [2]. Flow matching connects to continuous normalizing flows [11, 19, 39]. Optimal transport (OT) offers a metric view of distribution comparison [40, 52]. Computational OT methods include Sinkhorn divergence [13] and sliced Wasserstein distances [6,24]. OT-CFM uses OT to improve flow matching trajectories [41,47]. Our focus differs: transport costs serve not as a way to alter the probability path but as an auxiliary reliability signal that modulates backbone features.

Robustness in robot learning is a long-standing theme. Domain randomization broadens training support [46]. Domain adaptation aligns source and target distributions [51]. Invariance and out-of-distribution generalization are studied in broader machine learning [25]. Shortcut learning is a recurring failure mode [18]. Non-stationary and spurious correlations have been analyzed in robotics benchmarks [15, 55, 58]. This work complements these directions with a mechanism that reacts to drift at inference time through an internal transport-based signal.

## 3 Preliminaries

A VLA policy receives observations $o$, language instructions, and proprioceptive states. A vision-language backbone processes $o$ and produces context features $H = (h_1, \dots, h_T) \in \mathbb{R}^{T \times d}$, where $T$ is the sequence length and $d$ is the feature dimension. $H$ is the interface consumed by the action expert; depending on architecture, it can be the transformer hidden states or KV-cached activations.

The action head predicts an action chunk $a = (a_1, \dots, a_K) \in \mathbb{R}^{K \times d_a}$ with horizon $K$ and action dimension $d_a$. Flow matching generates $a$ by learning a conditional velocity field $v_\theta(x, t \mid H)$ over $x \in \mathbb{R}^{K \times d_a}$ and $t \in [0, 1]$. The training interpolant is $x_t = (1 - t)\,\varepsilon + t\,a$ with noise $\varepsilon \sim \mathcal{N}(0, I)$ and $t \sim \mathcal{U}[0, 1]$. The flow-matching objective regresses $v_\theta$ to the conditional target velocity [27]:

$$\mathcal{L}_{\mathrm{FM}}(\theta) \;=\; \mathbb{E}_{t,\varepsilon,a}\Big[\big\|v_\theta(x_t, t \mid H) - u(x_t, t, a)\big\|^2\Big]. \tag{1}$$

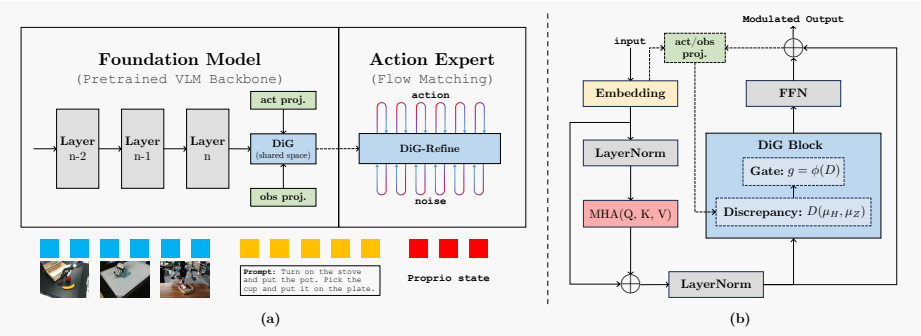



**Fig. 1: Overview of DiG.** The backbone produces observation features $H$. The action projection $f$ (the input projector from the action expert) maps the action chunk into the same feature space, yielding $Z$. A sliced Wasserstein cost computes the transport discrepancy $D$ between $H$ and $Z$. An exponential mapping converts $D$ into a gate $g$, which modulates a residual refinement to produce $\tilde{H}$. $\tilde{H}$ is then passed to the action expert for action generation. The same gate also reweights the training loss.

Optimal transport defines a cost between two measures $\mu, \nu$ over $\mathbb{R}^d$. The squared 2-Wasserstein distance is

$$W_2^2(\mu, \nu) = \inf_{\gamma \in \Gamma(\mu,\nu)} \int \|x - y\|^2 \, d\gamma(x, y), \tag{2}$$

where $\Gamma(\mu, \nu)$ is the set of couplings with marginals $\mu$ and $\nu$ [52]. Sliced Wasserstein distances approximate $W_2$ by averaging 1D Wasserstein costs along $M$ random projections on the unit sphere [6, 24]; $M$ controls the trade-off between variance and computation.

## 4 Discrepancy gating from transport reliability

This section introduces DiG as a transport-based reliability module. It also provides a theoretical perspective that motivates the gate mapping and clarifies when the mechanism improves robustness.

### 4.1 Transport discrepancy as an internal reliability signal

The action expert's input projection $f$ maps action chunks into the same feature space as the backbone output $H$. A transport cost between these two representations quantifies their compatibility: low cost indicates that $H$ lies in a region the action expert can decode reliably, while high cost signals drift away from that region. This cost serves as a per-step reliability estimate without extra supervision. During training, it reduces the influence of shortcut-like pairs. At test time, it flags when a predicted chunk is inconsistent with the current observation.

### 4.2 `DiG` module and training

Figure 1 gives an overview of `DiG`. The module attaches to the backbone context features consumed by the action expert. In a shared-transformer design, it refines the hidden states before the action expert. In a two-stage design, it refines the conditioning representation passed to a separate diffusion transformer.

**Action representation in the discrepancy space.** The transport discrepancy compares observation representations against an action representation expressed in the same feature coordinates. The action projection $f : \mathbb{R}^{d_a} \to \mathbb{R}^d$ is the input projection layer already present inside the action expert; it maps (noised) actions into the expert's internal feature dimension. Because $f$ sits on the main action-generation path, it receives gradients from the standard flow-matching loss without any additional training signal. `DiG` reuses $f$ directly: each action step yields $z_k = f(a_k) \in \mathbb{R}^d$, and a mean-pooled centroid $\bar{z} = \frac{1}{K}\sum_{k=1}^{K} z_k$ summarizes the chunk. The centroid is broadcast to length $T$ to form $Z = (\bar{z}, \ldots, \bar{z}) \in \mathbb{R}^{T\times d}$, removing action ordering effects and focusing the discrepancy on overall feature alignment. The centroid is used only inside the discrepancy branch; the flow-matching head still receives and models the full action chunk. This design keeps the reliability signal focused on chunk-level compatibility. Finer-grained variants, such as multi-centroid or temporal-bin discrepancies, can further distinguish temporal modes that share a similar average action.

**Transport discrepancy.** Empirical measures $\mu_H = \frac{1}{T}\sum_{i=1}^{T} \delta_{h_i}$ and $\mu_Z = \frac{1}{T}\sum_{i=1}^{T} \delta_{\bar{z}}$ are constructed from the feature sequences. The transport discrepancy is a sliced 2-Wasserstein cost

$$D(\mu_H, \mu_Z) \;=\; \frac{1}{M}\sum_{m=1}^{M} W_2^2\big(\langle\theta_m, \mu_H\rangle, \langle\theta_m, \mu_Z\rangle\big), \tag{3}$$

Because $\mu_Z$ is a point mass, each projected term admits a closed-form expression. For a unit direction $\theta$, define $u_i = \theta^\top h_i$ and $v = \theta^\top \bar{z}$. Then

$$D_\theta \;=\; W_2^2\big(\langle\theta, \mu_H\rangle, \langle\theta, \mu_Z\rangle\big) \;=\; \frac{1}{T}\sum_{i=1}^{T}(u_i - v)^2, \tag{4}$$

where $\langle\theta, \mu\rangle$ denotes the pushforward of $\mu$ under $x \mapsto \theta^\top x$. The supplementary material gives the full estimator.

**Gate mapping and temperature.** The discrepancy is mapped to a gate value

$$g \;=\; \phi(D) \;=\; \max\{g_{\min}, \exp(-\tau D)\}, \tag{5}$$

with clamp $g_{\min} \in (0, 1)$. The temperature $\tau > 0$ controls sensitivity: larger $\tau$ makes $g$ decay faster with $D$, while smaller $\tau$ yields a softer response. For training, we write $\bar{g} = \mathrm{sg}(g)$ for the scalar gate that enters the action-generation path; at inference $\bar{g} = g$. The forward value is unchanged, and the stop-gradient only blocks the path from the discrepancy score to the gate.

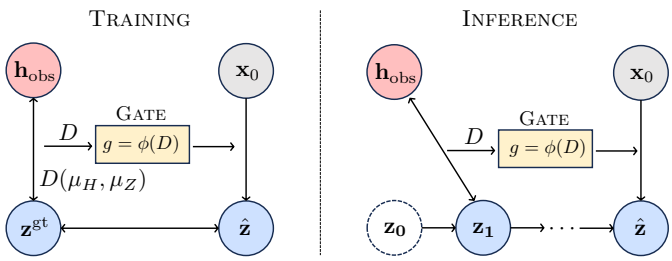



**Fig. 2: Training and `DiG-Refine`.** During training, the gate is computed from ground-truth actions; $\text{sg}(\cdot)$ marks the stop-gradient path, while $f$ remains on the normal action-generation path. During inference (`DiG-Refine`), the gate is computed from predicted actions and used inside the refinement loop.

**Gated residual refinement.** A residual operator $\mathcal{R}$ refines the context features:

$$\tilde{H} \;=\; H + \lambda\, \bar{g}\, \mathcal{R}(H), \tag{6}$$

with strength $\lambda > 0$. $\mathcal{R}$ is a lightweight linear map applied to the same normalized context features consumed by the action expert. This avoids assumptions about a specific normalization layer.

**Training objective and stop-gradient.** Training uses the ground-truth action chunk $a^{\text{gt}}$ to compute $Z$ and the gate value $g$. The flow-matching loss is reweighted as

$$J(\theta) \;=\; \mathbb{E}\Big[\bar{g}\, \mathcal{L}_{\text{FM}}(\theta; \tilde{H}, a^{\text{gt}})\Big]. \tag{7}$$

Here $\text{sg}(\cdot)$ is applied *only* to the gate, so $\bar{g}$ acts as a constant per-sample weight during backpropagation. Gradients still flow through $\mathcal{L}_{\text{FM}}$ and through $\tilde{H}$ in Eq. (6), training the backbone, $\mathcal{R}$, and the action expert (including $f$) normally. The target velocity $u(x_t, t, a)$ is unchanged; `DiG` changes sample weighting and the conditioned representation, not the flow-matching target. The stop-gradient blocks only the path $D \to g$, preventing the policy from shrinking the discrepancy or inflating $g$ without improving action prediction. Figure 2 summarizes the training and inference pathways.

### 4.3 Inference-time refinement with `DiG-Refine`

`DiG-Refine` is an inference-time procedure that reuses the same gate mapping and residual operator, but computes the action representation from predicted action chunks instead of ground-truth ones.

At control step $t$, the previous predicted chunk $\hat{a}_{t-1}$ is available. This breaks the circular dependency between the current gate and the current prediction. The current observation $o_t$ yields context features $H_t$. An initial gate is computed from the current features and the previous chunk:

$$g_t^{\text{prev}} \;=\; \phi\big(D(\mu_{H_t}, \mu_{Z(\hat{a}_{t-1})})\big)\,, \qquad H_t^{(0)} \;=\; H_t + \lambda\, g_t^{\text{prev}}\, \mathcal{R}(H_t), \tag{8}$$

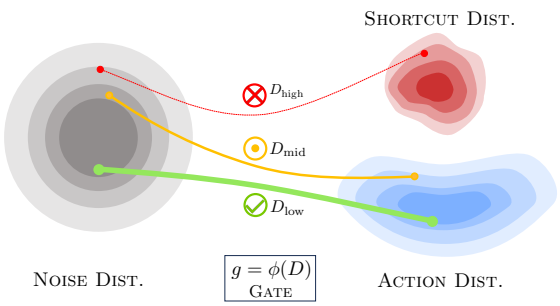



**Fig. 3: Transport discrepancy and the compatibility manifold.** The curved surface denotes the decodable region of representation space. Green points illustrate coherent observation-action pairs near this region, with low transport cost and high gate value. Red points illustrate shifted or shortcut-correlated pairs farther from this region, with high transport cost and low gate value.

and the action expert produces an initial chunk $\hat{a}_t^{(0)} = \text{FMHead}(H_t^{(0)})$. Further refinement uses the current prediction to update the chunk:

$$\hat{a}_t^{(i+1)} \;=\; \text{FMHead}\Big(H_t + \lambda\, \phi\Big(D(\mu_{H_t}, \mu_{Z(\hat{a}_t^{(i)})})\Big)\, \mathcal{R}(H_t)\Big)\,. \tag{9}$$

After $N$ refinements, the policy executes the first action in $\hat{a}_t^{(N)}$ and replans at the next control step. $N = 0$ yields the single-pass version that relies only on $g_t^{\text{prev}}$. The refinement is trust-gated: large discrepancy makes $g$ small and keeps the residual conservative, rather than requesting a larger correction from an unreliable action representation. Section 5.3 shows that a small number of rounds captures most gains.

### 4.4 Theoretical motivation

Two theoretical results motivate specific design choices in `DiG`. The first connects the exponential mapping in Eq. (5) to a local manifold noise model. The second formalizes how discrepancy-based gating suppresses shortcut components under a mixture contamination model (illustrated geometrically in Figure 3).

**Proposition 1 (Exponential gate as a posterior proxy).** *Consider one projected term $D_\theta$ in Eq. (4). Assume the projected residuals $u_i - v$ are i.i.d. $\mathcal{N}(0, \sigma^2)$. Then $D_\theta$ is a scaled chi-square random variable and its density satisfies*

$$p(D_\theta \mid \sigma^2) \;=\; C\, D_\theta^{T/2-1} \exp\!\left(-\frac{T}{2\sigma^2} D_\theta\right), \qquad D_\theta > 0, \tag{10}$$

*where $C$ is a constant independent of $D_\theta$.*

A compatibility-manifold view assumes that meaningful observation-action pairs concentrate near a decodable subset of the representation space. Drift

from this subset occurs mainly along normal directions. Modeling such drift as Gaussian noise, Eq. (10) shows that larger discrepancy makes the on-manifold explanation exponentially less likely. The map $g{=}\exp(-\tau D)$ in Eq. (5) turns this likelihood into a gate value, with $\tau$ controlling how aggressively high-discrepancy cases are downweighted.

Now consider a mixture model for training pairs $(H, a)$. Let $P_{\mathrm{coh}}$ denote a coherent distribution and $P_{\mathrm{sh}}$ denote a shortcut distribution. The training distribution is

$$P \;=\; (1-\rho)\,P_{\mathrm{coh}} + \rho\,P_{\mathrm{sh}}, \qquad \rho \in (0,1). \tag{11}$$

The gate $g(H, a) \in [g_{\min}, 1]$ induces a reweighted distribution

$$Q(dH\,da) \;\propto\; g(H,a)\,P(dH\,da). \tag{12}$$

**Theorem 1 (Gating increases coherent mass).** *Assume the mixture model in Eq. (11). Consider an idealized* component-wise *gate that is constant on each mixture component. It takes value $g_1$ on samples drawn from $P_{\mathrm{coh}}$ and value $g_0$ on samples drawn from $P_{\mathrm{sh}}$, with $g_1 > g_0 > 0$. Define $Q$ via Eq. (12). Then $Q$ is a mixture of the same components,*

$$Q \;=\; (1-\rho_Q)\,P_{\mathrm{coh}} + \rho_Q\,P_{\mathrm{sh}}, \tag{13}$$

*with shortcut weight*

$$\rho_Q \;=\; \frac{\rho\,g_0}{(1-\rho)\,g_1 + \rho\,g_0} \;<\; \rho. \tag{14}$$

*Moreover, for any loss $\ell \in [0,1]$ and any parameter $\theta$,*

$$\left|\,\mathbb{E}_{(H,a)\sim Q}[\ell(\theta)] - \mathbb{E}_{(H,a)\sim P_{\mathrm{coh}}}[\ell(\theta)]\,\right| \;\le\; \rho_Q. \tag{15}$$

Eq. (14) makes the mechanism explicit. When the gate is larger on the coherent component than on the shortcut component, importance reweighting reduces the shortcut fraction in the induced distribution $Q$. Eq. (15) is a worst-case bound that does not assume structure of the losses. In practice, $g$ is a continuous function of the transport discrepancy. The design goal is to assign larger gate values to pairs that stay near the decodable manifold and smaller values to pairs that drift away. Proposition 1 provides a local manifold model where squared deviations produce an exponential likelihood in $D$, which motivates the exponential mapping in Eq. (5). Proofs are provided in the supplementary material.

## 5 Experiments

Experiments evaluate `DiG` along three axes: overall performance on simulation and real-robot benchmarks, robustness under distribution shift, and the contribution of each design component.

Two backbones are used to test compatibility with different architectures. The first is $\pi_{0.5}$ [21] with an action expert coupled to a large vision-language backbone. The second is GR00T-N1 [4], a dual-system VLA whose DiT action expert is conditioned on vision-language token embeddings. In both cases, `DiG` attaches at the representation interface consumed by the action expert.

**Table 1: Success rates (%) on LIBERO.** Mean success over 50 rollouts per task.

| Method | Spatial | Object | Goal | Long | Avg |
|---|---|---|---|---|---|
| Diffusion Policy [12] | 78.5 | 87.5 | 73.5 | 64.8 | 76.1 |
| SpatialVLA [42] | 88.2 | 89.9 | 78.6 | 55.5 | 78.1 |
| CoT-VLA [63] | 87.5 | 91.6 | 87.6 | 69.0 | 83.9 |
| OpenVLA [23] | 84.7 | 88.4 | 79.2 | 53.7 | 76.5 |
| OpenVLA-OFT [22] | 97.6 | 98.4 | 97.9 | 94.5 | 97.1 |
| GR00T-N1 [4] | 94.4 | 97.6 | 93.0 | 90.6 | 93.9 |
| $\pi_0$ [5] | 98.0 | 96.8 | 94.4 | 88.4 | 94.4 |
| $\pi_{0.5}$ [21] | 98.8 | 98.2 | 98.0 | 92.4 | 96.9 |
| GR00T-N1+`DiG` (Ours) | 96.0 | 98.4 | 94.8 | 92.1 | 95.3 |
| $\pi_{0.5}$+`DiG` (Ours) | **99.2** | **99.0** | **98.6** | **96.4** | **98.3** |

**Table 2: RoboCasa 24-task benchmark with 50 demonstrations per task.** Success rates (%) averaged within categories over 50 rollouts per task.

| Method | Pick & Place | Doors/Drawers | Others | Avg |
|---|---|---|---|---|
| GR00T-N1 | 18.6 | 50.2 | 39.1 | 36.0 |
| GR00T-N1+`DiG` (Ours) | 22.4 | 60.3 | 47.0 | **43.2** |
| $\pi_{0.5}$ | 21.5 | 57.8 | 44.9 | 41.4 |
| $\pi_{0.5}$+`DiG` (Ours) | 27.2 | 73.4 | 57.2 | **52.6** |

### 5.1 Simulation benchmarks

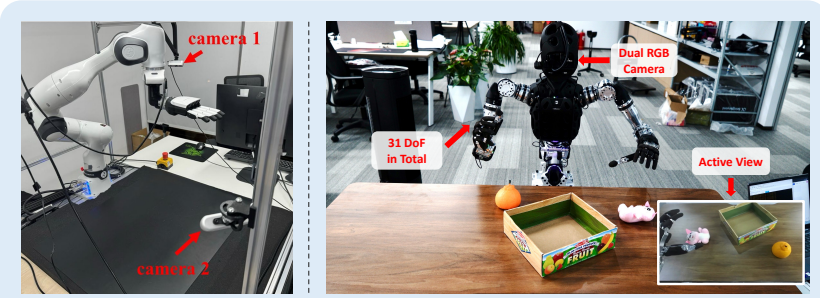
Two simulation benchmarks are used. **LIBERO** [28] is a 40-task suite for tabletop manipulation conditioned on language instructions, organized into four categories of increasing difficulty (Spatial, Object, Goal, Long). **RoboCasa** [38] is a photorealistic household simulator with 24 kitchen tasks spanning pick-and-place, door/drawer manipulation, and multi-step cooking. RoboCasa is evaluated in a low-data regime with 50 demonstrations per task to stress-test generalization under limited supervision.

**Main results on LIBERO.** Table 1 compares `DiG` against recent VLA baselines. `DiG` improves both backbones and achieves the highest average success rate. The gains are not uniform across task categories: LIBERO-Spatial and LIBERO-Object, where the baseline already exceeds 98%, see modest improvements (+0.4 to +0.8), while LIBERO-Long improves by +4.0 with $\pi_{0.5}$. This asymmetry is expected: long-horizon tasks involve more control steps, so early representational errors compound and the reliability gate has more opportunities to intervene. On shorter tasks where the baseline is near ceiling, the gate values stay high and `DiG` reduces to near-identity refinement. GR00T-N1+`DiG` also improves over the GR00T-N1 baseline (+1.4 avg), confirming that the module generalizes across architectures with different action-head designs.

**Few-shot results on RoboCasa.** Table 2 shows that `DiG` yields larger absolute gains here than on LIBERO (+11.2 avg for $\pi_{0.5}$, +7.2 for GR00T-N1). With scarce demonstrations, the backbone is more prone to overfitting to spurious correlations in the training set. The transport discrepancy acts as a soft regularizer:

**Table 3: Robustness to non-stationary perturbations.** Success rates (%) under time-varying noise. Numbers in color show change relative to the baseline row.

| Perturbation | Method | Spatial | Object | Goal | Long | Avg |
|---|---|---|---|---|---|---|
| Cosine only | w/o DiG | **87.6** | **89.8** | 86.4 | 70.2 | 83.5 |
| | w/ DiG | 87.4 -0.2 | 89.8 0.0 | **91.6** +5.2 | **80.0** +9.8 | **87.2** +3.7 |
| Sine only | w/o DiG | 88.2 | 88.4 | 87.2 | 68.6 | 83.1 |
| | w/ DiG | **90.0** +1.8 | **90.6** +2.2 | **91.2** +4.0 | **80.4** +11.8 | **88.1** +5.0 |
| Cosine + sine | w/o DiG | **87.8** | 87.2 | 86.0 | 67.8 | 82.2 |
| | w/ DiG | 86.4 -1.4 | **88.2** +1.0 | **90.8** +4.8 | **79.2** +11.4 | **86.2** +4.0 |




(a) FR3+Inspire Hand (b) PND Adam-U

**Fig. 4: (a) FR3 + Inspire Hand Setup.** Two RGB cameras and a dexterous hand. **(b) PND Adam-U Setup.** A 31-DoF humanoid with dual dexterous hands and an actively controlled head mounting dual RGB cameras.

samples whose observation features are poorly aligned with the action representation receive lower gate values, reducing their influence on the loss. The effect is strongest on Pick & Place (+5.7 for $\pi_{0.5}$) and Doors/Drawers (+15.6), categories that involve spatial alignment where spurious visual cues are most harmful.

**Non-stationary perturbations.** Table 3 adds time-varying noise (cosine, sine, or both) to observations and proprioception following the protocol of [15, 58]. A consistent pattern emerges across all perturbation types: DiG improves substantially on hard tasks like Goal and Long (+4.0 to +11.8). Spatial tasks are short and the perturbation has limited time to accumulate drift. Goal and Long tasks require sustained multi-step execution where time-varying noise compounds through the chunked control loop.

### 5.2 Real robot benchmarks

Simulation results establish that DiG improves performance in controlled settings. Real-robot evaluation tests whether these gains persist under real sensor noise, contact dynamics, and human intervention. Two platforms are used (Figure 4): (i) a 7-DoF Franka FR3 arm equipped with an Inspire dexterous hand, observed by two fixed RGB cameras, evaluated on four tabletop manipulation tasks with 20 trials per task; (ii) a PND Adam-U humanoid upper body (31

**Table 4: Franka real-robot success rates (%).** Each cell reports sub-task / whole-task success over 20 trials.

| | Method | Stack-Bowls | Spray-Plant | Wipe-Whiteboard | Sort-Into-Drawer |
|---|---|---|---|---|---|
| **In-dist** | $\pi_{0.5}$ | 60 / 40 | 65 / 45 | 60 / 40 | 50 / 35 |
| | $\pi_{0.5}$+DiG (Ours) | **65** / **50** | **70** / **50** | **70** / **50** | **60** / **40** |
| **OOD** | $\pi_{0.5}$ | 40 / 15 | 35 / 10 | 45 / 20 | 50 / 20 |
| | $\pi_{0.5}$+DiG (Ours) | **65** / **40** | **60** / **30** | **60** / **30** | **60** / **35** |

**(a)** Unseen object/background **(b)** Human intervention

**Fig. 5: Perturbations in the OOD setting.** Background textures are changed and human intervention is introduced during execution.

DoF) with dual dexterous hands and an actively controlled head mounting dual RGB cameras, evaluated on an active-view clean-up task with 20 trials per configuration. Both platforms use $\pi_{0.5}$ as the backbone. Further hardware and task details are provided in the supplementary material.

**Franka tasks.** Table 4 reports sub-task and whole-task success rates across four tasks. In the in-distribution setting, DiG improves all tasks, with whole-task gains ranging from +5 (Spray-Plant, Sort-Into-Drawer) to +10 (Stack-Bowls, Wipe-Whiteboard). Under OOD conditions (background texture changes and human intervention, visualized in Figure 5), the gap widens substantially. The baseline whole-task success drops to 10% to 20%, while DiG maintains 30% to 40%. The largest relative improvement appears on Spray-Plant under OOD (+25 sub-task, +20 whole-task), a tool-use task where visual distractors can cause the policy to mislocate the bottle. Sort-Into-Drawer, the longest-horizon task, shows the smallest in-distribution gap, yet DiG still provides meaningful improvement under OOD, consistent with the pattern observed in simulation.

**Adam-U active-view tasks.** Table 5 evaluates a 31-DoF humanoid on an active-view clean-up task with 1 to 3 objects under seen and unseen backgrounds (Figure 6). In the seen setting, DiG matches or improves the baseline, with gains growing as the number of objects increases (+0 for 1 object, +10 for 3 objects). More objects require longer execution and more head-camera viewpoint changes, amplifying the chance of representational drift. Under unseen backgrounds, the baseline drops sharply (75 $\rightarrow$ 55 for 1 object, 35 $\rightarrow$ 25 for 3 objects). DiG recovers a substantial fraction: the 3-object unseen success rate improves from 25% to 40%, a +15 absolute gain. This confirms that the transport discrepancy captures background-induced drift that is invisible to the flow-matching loss alone.

**Table 5: Adam-U and visual-shift real-task success rates (%).** The left block reports active-view clean-up with 1 to 3 objects under seen and unseen backgrounds, 20 trials each. The right block reports visual-shift tasks under standard (Std) and OOD settings, 20 trials per cell; AF: Arrange-Flower, BH: Bimanual Handover.

| Method | Adam-U active-view clean-up | | | | | | Visual-shift real tasks | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Seen | | | Unseen | | | Std | | OOD | |
| | 1 obj | 2 objs | 3 objs | 1 obj | 2 objs | 3 objs | AF | BH | AF | BH |
| $\pi_{0.5}$ | 75 | 60 | 35 | 55 | 30 | 25 | 15 | 35 | 5 | 15 |
| $\pi_{0.5}$+DiG (Ours) | **75** | **65** | **45** | **65** | **45** | **40** | **35** | **55** | **25** | **40** |

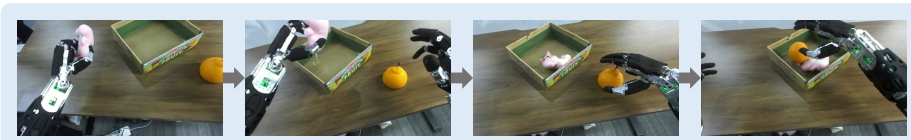


**Fig. 6: Adam-U active-view evaluation.** The head camera is actively controlled during execution. The seen and unseen settings differ in background and lighting.

**Visual-shift real tasks.** The right block of Table 5 and Figure 7 evaluate two real tasks that keep the task semantics fixed while changing backgrounds and lighting. Arrange-Flower stresses object rearrangement under cluttered visual context, and Bimanual Handover stresses coordinated dual-arm motion under the same appearance shift. DiG improves both standard and shifted variants, with larger gains in the OOD setting.

### 5.3 Ablations and analysis

**Discrepancy choice.** The metric-ablation columns in Table 6 vary the discrepancy while holding the gate form fixed at $g = \exp(-\tau D)$. All transport-based metrics outperform cosine distance and MMD, with sliced Wasserstein achieving the best average. The gap is most visible on LIBERO-Long ($-3.8$ for cosine, $-3.0$

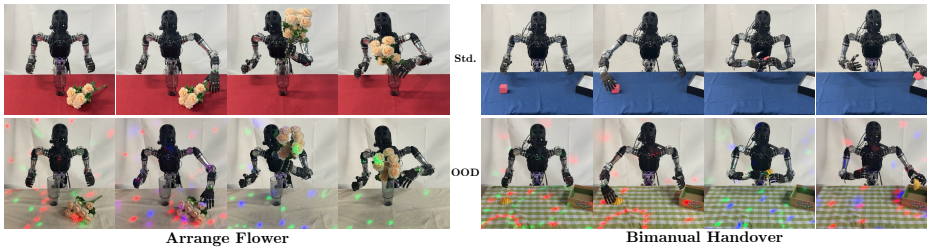


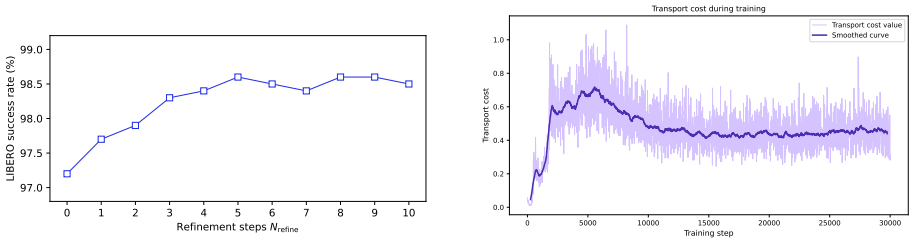

**Fig. 7: Real visual-shift settings.** Standard and OOD views for Arrange-Flower and Bimanual Handover.

**Table 6: Ablations on LIBERO (success %).** Metric columns vary the discrepancy with gate form fixed as $g = \exp(-\tau D)$. Gate columns vary the gate with discrepancy fixed as sliced Wasserstein. Red numbers denote drops from Ours.

| Task | Metric ablation | | | Gate ablation | | | | Ours |
|---|---|---|---|---|---|---|---|---|
| | Cos. | MMD | Sink. | Fix. | Rand. | $\lambda = 0$ | MLP | |
| **Spatial** | 98.2 -1.0 | 98.4 -0.8 | 98.8 -0.4 | 96.8 -2.4 | 95.2 -4.0 | 98.6 -0.6 | 98.8 -0.4 | **99.2** |
| **Object** | 97.8 -1.2 | 98.2 -0.8 | 98.6 -0.4 | 95.4 -3.6 | 93.8 -5.2 | 98.4 -0.6 | 98.4 -0.6 | **99.0** |
| **Goal** | 97.2 -1.4 | 97.6 -1.0 | 98.2 -0.4 | 92.2 -6.4 | 89.4 -9.2 | 98.0 -0.6 | 98.0 -0.6 | **98.6** |
| **Long** | 92.6 -3.8 | 93.4 -3.0 | 95.8 -0.6 | 84.6 -11.8 | 80.8 -15.6 | 94.2 -2.2 | 94.8 -1.6 | **96.4** |
| **Avg** | 96.5 -1.8 | 96.8 -1.5 | 97.9 -0.4 | 92.3 -6.0 | 89.8 -8.5 | 97.3 -1.0 | 97.5 -0.8 | **98.3** |

for MMD, $-0.6$ for Sinkhorn vs. sliced Wasserstein). Cosine distance collapses the distributional structure into a single scalar similarity, losing information about how the feature distribution spreads around the action centroid. MMD with an RBF kernel captures distributional differences but is sensitive to bandwidth selection; a single bandwidth cannot simultaneously resolve fine-grained structure in high-density regions and coarse drift in the tails. Sinkhorn is closest to sliced Wasserstein in performance, which is expected since both are transport-based; the sliced variant is cheaper to compute and avoids the entropic regularization bias. The performance gap between metrics widens on longer-horizon tasks. On Spatial and Object, where episodes are short and the baseline is near ceiling, all four metrics produce similar gate distributions and the residual correction has limited room to help. On Long, representational drift accumulates over many control steps, and the gate must distinguish between mild and severe misalignment. Transport-based metrics preserve this graded information, while cosine distance saturates early and MMD bandwidth mismatches blur the distinction.

**Gate mechanism.** The gate-ablation columns in Table 6 vary the gate while holding the discrepancy fixed at sliced Wasserstein. A fixed gate ($g = 0.5$) applies the same residual correction regardless of the input, removing all discrepancy-based modulation. This ablation still benefits from the residual operator $\mathcal{R}$ but loses the ability to adapt per sample. The $-6.0$ avg drop (and $-11.8$ on Long) confirms that discrepancy-driven gating is the primary source of improvement, not the residual correction alone. A random gate performs even worse ($-8.5$ avg), since random modulation introduces noise into the refinement and the loss reweighting. Two targeted variants separate the roles of loss reweighting, residual refinement, and discrepancy estimation. Setting $\lambda = 0$ removes the residual branch but keeps loss reweighting, so the reliability signal alone improves over the baseline while remaining below full `DiG`. The MLP gate replaces the transport discrepancy with a learned scalar gate on pooled $H/\bar{z}$ while keeping the same residual and loss paths; its smaller gain indicates that the geometric discrepancy is more informative than an unconstrained learned gate. The targeted variants close part of the gap but stay below full `DiG`, while the non-adaptive controls follow fixed gate > random gate. The transport gate adds sample-adaptive modulation on top, and its advantage grows with task horizon, confirming that per-step reliability estimation is most valuable when errors compound.




**(a) Refinement iterations.** Success rate vs. refinement steps.

**(b) Training discrepancy.** Raw and smoothed cost vs. training steps.

**Fig. 8: Ablation diagnostics.**

**Refinement iterations.** `DiG-Refine` trades compute for robustness. Figure 8a shows that performance improves sharply from $N = 0$ (single-pass, using only the previous chunk's gate) to $N = 1$, then continues to rise modestly until $N = 3$, after which it saturates. The initial jump from $N = 0$ to $N = 1$ reflects the value of using the current prediction to recompute the gate: the previous chunk may be stale, especially after large observation changes. Saturation after $N = 3$ suggests that the refinement loop reaches a stable self-consistent chunk quickly; later iterations have little room to change the chunk once the gate and action representation agree. In terms of latency, each refinement iteration requires one additional forward pass through the action expert but not through the vision-language backbone, since $H_t$ and $\mathcal{R}(H_t)$ are cached after the first pass. On a single A100 GPU, one iteration adds approximately 8 ms for $\pi_{0.5}$, so $N = 3$ increases per-step inference time by roughly 24 ms, well within the 100 ms budget of a 10 Hz control loop.

**Transport discrepancy stays informative.** A potential failure mode is discrepancy collapse: if $D$ converges to a constant, the gate saturates and stops reflecting reliability. Figure 8b plots the average transport discrepancy during training. The curve decreases as the backbone learns better representations but stabilizes at a non-trivial level rather than collapsing to zero. This confirms that the stop-gradient on the gate prevents the model from directly minimizing $D$, preserving its diagnostic value throughout training. The supplementary material further reports per-step reliability diagnostics.

**Hyperparameter sensitivity.** Figure 9 reports sensitivity to the number of projections $M$ and to $(\lambda, \tau)$. Performance is stable across a wide range of $M$: even $M = 16$ projections yield most of the gain, and increasing beyond $M = 32$ provides diminishing returns. This is consistent with known convergence properties of sliced Wasserstein estimators. The $(\lambda, \tau)$ sweep shows that moderate values ($\lambda \approx 0.1$, $\tau \approx 1.0$) work well. Very large $\lambda$ over-amplifies the residual correction and can destabilize training. Very small $\tau$ makes the gate nearly constant across samples, reducing `DiG` to a uniform residual correction similar to the fixed-gate ablation. The gate clamp $g_{\min} = 0.05$ prevents complete suppression of any training sample. Setting $g_{\min} = 0$ allows the gate to fully zero

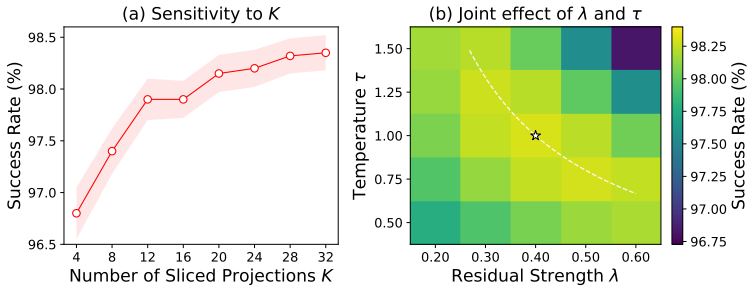



**Fig. 9: Hyperparameter sensitivity.** Left: varying the number of sliced projections $M$. Right: joint sweep of refinement strength $\lambda$ and temperature $\tau$.

out high-discrepancy pairs, which risks discarding rare but valid demonstrations that happen to have unusual observation-action alignment. Conversely, raising $g_{\min}$ above 0.2 weakens the reweighting effect and narrows the gap between coherent and shortcut samples, reducing the benefit of gating. The default value $g_{\min} = 0.05$ balances these two extremes and performs consistently across all benchmarks without per-task tuning.

## 6 Conclusion

Transport discrepancy provides a simple reliability signal for VLA policies that use flow matching in the action expert. `DiG` uses this signal to gate a lightweight residual refinement and to reweight the flow-matching loss via $\mathrm{sg}(\cdot)$ applied only to the gate, so the signal stays diagnostic while learning still follows the flow-matching objective. `DiG-Refine` reuses the same mechanism at test time. Results across simulation and real robots show consistent robustness gains, with the largest improvements under strong shifts and long horizons.

The current evaluation covers two flow-matching VLA architectures. Extending `DiG` to discrete-action or 3D-based VLA models is a promising direction, as the transport discrepancy concept applies whenever observation and action representations share a common feature space. The theoretical analysis relies on a simplified mixture model with Gaussian noise; characterizing behavior under more general distribution shifts remains an open direction. On the practical side, `DiG-Refine` adds $N$ extra action-expert forward passes per step, which may matter for control loops faster than 10 Hz. The temperature $\tau$ and projection count $M$ are currently tuned per backbone; automatic calibration would ease transfer to new domains.

## Acknowledgements

This work was supported by NSFC in part under Grant 62450001 and 62476008. The authors would like to thank the anonymous reviewers for their valuable comments and advice.

## References


1. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. Advances in neural information processing systems **35**, 23716–23736 (2022)
2. Albergo, M.S., Boffi, N.M., Vanden-Eijnden, E.: Stochastic interpolants: A unifying framework for flows and diffusions. arXiv preprint arXiv:2303.08797 (2023)
3. Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., et al.: Qwen technical report. arXiv preprint arXiv:2309.16609 (2023)
4. Bjorck, J., Castañeda, F., Cherniaiev, N., Da, X., Ding, R., Fan, L., Fang, Y., Fox, D., Hu, F., Huang, S., et al.: Gr00t n1: An open foundation model for generalist humanoid robots. arXiv preprint arXiv:2503.14734 (2025)
5. Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., et al.: $\pi_0$: A vision-language-action flow model for general robot control. arXiv preprint arXiv:2410.24164 (2024)
6. Bonneel, N., Rabin, J., Peyré, G., Pfister, H.: Sliced and radon wasserstein barycenters of measures. In: Journal of Mathematical Imaging and Vision. vol. 51, pp. 22–45 (2015)
7. Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., Dubey, A., Finn, C., Florence, P., Fu, C., Arenas, M.G., Gopalakrishnan, K., Han, K., Hausman, K., Herzog, A., Hsu, J., Ichter, B., Irpan, A., Joshi, N., Julian, R., Kalashnikov, D., Kuang, Y., Leal, I., Lee, L., Lee, T.W.E., Levine, S., Lu, Y., Michalewski, H., Mordatch, I., Pertsch, K., Rao, K., Reymann, K., Ryoo, M., Salazar, G., Sanketi, P., Sermanet, P., Singh, J., Singh, A., Soricut, R., Tran, H., Vanhoucke, V., Vuong, Q., Wahid, A., Welker, S., Wohlhart, P., Wu, J., Xia, F., Xiao, T., Xu, P., Xu, S., Yu, T., Zitkovich, B.: Rt-2: Vision-language-action models transfer web knowledge to robotic control. arXiv preprint arXiv:2307.15818 (2023)
8. Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jackson, T., Jesmonth, S., Joshi, N.J., Julian, R., Kalashnikov, D., Kuang, Y., Leal, I., Lee, K.H., Levine, S., Lu, Y., Malla, U., Manjunath, D., Mordatch, I., Nachum, O., Parada, C., Peralta, J., Perez, E., Pertsch, K., Quiambao, J., Rao, K., Ryoo, M., Salazar, G., Sanketi, P., Sayed, K., Singh, J., Sontakke, S., Stone, A., Tan, C., Tran, H., Vanhoucke, V., Vega, S., Vuong, Q., Xia, F., Xiao, T., Xu, P., Xu, S., Yu, T., Zitkovich, B.: Rt-1: Robotics transformer for real-world control at scale. arXiv preprint arXiv:2212.06817 (2023)
9. Cheang, C.L., Chen, G., Jing, Y., Kong, T., Li, H., Li, Y., Liu, Y., Wu, H., Xu, J., Yang, Y., Zhang, H., Zhu, M.: Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. arXiv preprint arXiv:2410.06158 (2024)
10. Cheang, C., Chen, S., Cui, Z., Hu, Y., Huang, L., Kong, T., Li, H., Li, Y., Liu, Y., Ma, X., et al.: Gr-3 technical report. arXiv preprint arXiv:2507.15493 (2025)
11. Chen, R.T., Rubanova, Y., Bettencourt, J., Duvenaud, D.K.: Neural ordinary differential equations. In: Advances in neural information processing systems. vol. 31 (2018)
12. Chi, C., Xu, Z., Feng, S., Cousineau, E., Du, Y., Burchfiel, B., Tedrake, R., Song, S.: Diffusion policy: Visuomotor policy learning via action diffusion. arXiv preprint arXiv:2303.04137 (2023)

13. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. Advances in neural information processing systems **26** (2013)
14. Driess, D., Xia, F., Sajjadi, M.S., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., et al.: Palm-e: An embodied multimodal language model. In: International Conference on Machine Learning. pp. 8469–8488. PMLR (2023)
15. Feng, F., Huang, B., Zhang, K., Magliacane, S.: Factored adaptation for non-stationary reinforcement learning. Advances in Neural Information Processing Systems **35**, 31957–31971 (2022)
16. Feng, Y., Li, Y., Zhang, W., Zheng, S., Luo, H., Yue, Z., Lu, Z.: Videoorion: Tokenizing object dynamics in videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 20401–20412 (October 2025)
17. Feng, Y., Zhang, W., Wang, Y., Luo, H., Yuan, H., Zheng, S., Lu, Z.: Spatial-aware vla pretraining through visual-physical alignment from human videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2026)
18. Geirhos, R., Jacobsen, J.H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., Wichmann, F.A.: Shortcut learning in deep neural networks. Nature Machine Intelligence **2**(11), 665–673 (2020)
19. Grathwohl, W., Chen, R.T., Bettencourt, J., Sutskever, I., Duvenaud, D.: Ffjord: Free-form continuous dynamics for scalable reversible generative models. arXiv preprint arXiv:1810.01367 (2018)
20. Huang, H., Liu, F., Fu, L., Wu, T., Mukadam, M., Malik, J., Goldberg, K., Abbeel, P.: Otter: A vision-language-action model with text-aware visual feature extraction. arXiv preprint arXiv:2503.03734 (2025)
21. Intelligence, P., Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., et al.: $\pi_{0.5}$: a vision-language-action model with open-world generalization. arXiv preprint arXiv:2504.16054 (2025)
22. Kim, M.J., Finn, C., Liang, P.: Fine-tuning vision-language-action models: Optimizing speed and success. arXiv preprint arXiv:2502.19645 (2025)
23. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Baljekar, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., et al.: Openvla: An open-source vision-language-action model. arXiv preprint arXiv:2406.09246 (2024)
24. Kolouri, S., Nadjahi, K., Simsekli, U., Badeau, R., Rohde, G.: Generalized sliced wasserstein distances. Advances in neural information processing systems **32** (2019)
25. Krueger, D., Caballero, E., Jacobsen, J.H., Zhang, A., Binas, J., Zhang, D., Le Priol, R., Courville, A.: Out-of-distribution generalization via risk extrapolation (rex). In: International conference on machine learning. pp. 5815–5826. PMLR (2021)
26. Lin, F., Nai, R., Hu, Y., You, J., Zhao, J., Gao, Y.: Onetwovla: A unified vision-language-action model with adaptive reasoning. arXiv preprint arXiv:2505.11917 (2025)
27. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2023)
28. Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., Stone, P.: Libero: Benchmarking knowledge transfer for lifelong robot learning. arXiv preprint arXiv:2306.03310 (2023)
29. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 26296–26306 (2024)

30. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems **36**, 34892–34916 (2023)
31. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. arXiv preprint arXiv:2209.03003 (2022)
32. Lu, Y., Li, C., Liu, H., Yang, J., Gao, J., Shen, Y.: An empirical study of scaling instruct-tuned large multimodal models. arXiv preprint arXiv:2309.09958 (2023)
33. Luo, H., Feng, Y., Zhang, W., Zheng, S., Wang, Y., Yuan, H., Liu, J., Xu, C., Jin, Q., Lu, Z.: Being-H0: Vision-language-action pretraining from large-scale human videos. In: International Conference on Machine Learning. PMLR (2026)
34. Luo, H., Wang, Y., Zhang, W., Yuan, H., Feng, Y., Xu, H., Zheng, S., Lu, Z.: Joint-aligned latent action: Towards scalable vla pretraining in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2026)
35. Luo, H., Wang, Y., Zhang, W., Zheng, S., Xi, Z., Xu, C., Xu, H., Yuan, H., Zhang, C., Wang, Y., et al.: Being-H0.5: Scaling human-centric robot learning for cross-embodiment generalization. arXiv preprint arXiv:2601.12993 (2026)
36. Luo, H., Yue, Z., Zhang, W., Feng, Y., Zheng, S., Ye, D., Lu, Z.: OpenMMEgo: Enhancing egocentric understanding for LMMs with open weights and data. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025)
37. Luo, H., Zhang, W., Feng, Y., Zheng, S., Xu, H., Xu, C., Xi, Z., Fu, Y., Lu, Z.: Being-H0.7: A latent world-action model from egocentric videos. arXiv preprint arXiv:2605.00078 (2026)
38. Nasiriany, S., Maddukuri, A., Zhang, L., Parikh, A., Lo, A., Joshi, A., Mandlekar, A., Zhu, Y.: Robocasa: Large-scale simulation of everyday tasks for generalist robots. arXiv preprint arXiv:2406.02523 (2024)
39. Papamakarios, G., Nalisnick, E., Rezende, D.J., Mohamed, S., Lakshminarayanan, B.: Normalizing flows for probabilistic modeling and inference. Journal of Machine Learning Research **22**(57), 1–64 (2021)
40. Peyré, G., Cuturi, M.: Computational optimal transport. Foundations and Trends in Machine Learning **11**(5-6), 355–607 (2019)
41. Pooladian, A.A., Ben-Hamu, H., Domingo-Enrich, C., Amos, B., Lipman, Y., Chen, R.T.: Multisample flow matching: Straightening flows with minibatch couplings. arXiv preprint arXiv:2304.14772 (2023)
42. Qu, D., Song, H., Chen, Q., Yao, Y., Ye, X., Ding, Y., Wang, Z., Gu, J., Zhao, B., Wang, D., Li, X.: Spatialvla: Exploring spatial representations for visual-language-action model. arXiv preprint arXiv:2501.15830 (2025)
43. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
44. Sun, J., Zhang, W., Qi, Z., Ren, S., Liu, Z., Zhu, H., Sun, G., Jin, X., Chen, Z.: Vla-jepa: Enhancing vision-language-action model with latent world model. arXiv preprint arXiv:2602.10098 (2026)
45. Team, O.M., Ghosh, D., Walke, H., Pertsch, K., Black, K., Mees, O., Dasari, S., Hejna, J., Kreiman, T., Xu, C., Luo, J., Tan, Y.L., Chen, L.Y., Sanketi, P., Vuong, Q., Xiao, T., Sadigh, D., Finn, C., Levine, S.: Octo: An open-source generalist robot policy. arXiv preprint arXiv:2405.12213 (2024)
46. Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P.: Domain randomization for transferring deep neural networks from simulation to the real

world. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS). pp. 23–30. IEEE (2017)
47. Tong, A., Fatras, K., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Wolf, G., Bengio, Y.: Improving and generalizing flow-based generative models with minibatch optimal transport. arXiv preprint arXiv:2302.00482 (2024)
48. Tong, A., Huang, J., Wolf, G., Van Dijk, D., Krishnaswamy, S.: Conditional flow matching: Simulation-free training of continuous normalizing flows. arXiv preprint arXiv:2302.00482 (2023)
49. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
50. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
51. Tzeng, E., Devin, C., Hoffman, J., Finn, C., Abbeel, P., Levine, S., Saenko, K., Darrell, T.: Adapting deep visuomotor representations with weak pairwise constraints. In: Algorithmic Foundations of Robotics XII: Proceedings of the Twelfth Workshop on the Algorithmic Foundations of Robotics. pp. 688–703. Springer (2020)
52. Villani, C.: Optimal transport: old and new, vol. 338. Springer (2009)
53. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., et al.: Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. arXiv preprint arXiv:2409.12191 (2024)
54. Wang, Y., Zheng, S., Luo, H., Zhang, W., Yuan, H., Xu, C., Xu, H., Feng, Y., Yu, M., Kang, Z., et al.: Rethinking visual-language-action model scaling: Alignment, mixture, and regularization. arXiv preprint arXiv:2602.09722 (2026)
55. Xing, Y., Luo, X., Xie, J., Gao, L., Shen, H., Song, J.: Shortcut learning in generalist robot policies: The role of dataset diversity and fragmentation. arXiv preprint arXiv:2508.06426 (2025)
56. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 11975–11986 (2023)
57. Zhang, W., Feng, Y., Luo, H., Li, Y., Yue, Z., Zheng, S., Lu, Z.: Unified multimodal understanding via byte-pair visual encoding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 12976–12986 (October 2025)
58. Zhang, W., Li, Y., Yang, B., Lu, Z.: Tackling non-stationarity in reinforcement learning via causal-origin representation. In: Proceedings of the 41st International Conference on Machine Learning. vol. 235, pp. 59264–59288. PMLR (2024)
59. Zhang, W., Luo, H., Zheng, S., Feng, Y., Xu, H., Xi, Z., Xu, C., Yuan, H., Lu, Z.: Conservative offline robot policy learning via posterior-transition reweighting. arXiv preprint arXiv:2603.16542 (2026)
60. Zhang, W., Xie, Z., Feng, Y., Li, Y., Xing, X., Zheng, S., Lu, Z.: From pixels to tokens: Byte-pair encoding on quantized visual modalities. In: The Thirteenth International Conference on Learning Representations (2025)
61. Zhang, W., Liu, H., Qi, Z., Wang, Y., Yu, X., Zhang, J., Dong, R., He, J., Wang, H., Zhang, Z., et al.: Dreamvla: A vision-language-action model dreamed with comprehensive world knowledge. arXiv preprint (2025)
62. Zhang, W., Zhang, B., Qi, Z., Zeng, W., Jin, X., Zhang, L.: Disentangled robot learning via separate forward and inverse dynamics pretraining. arXiv preprint arXiv:2604.16391 (2026)

63. Zhao, Q., Lu, Y., Kim, M.J., Fu, Z., Zhang, Z., Wu, Y., Li, Z., Ma, Q., Han, S., Finn, C., Handa, A., Liu, M.Y., Xiang, D., Wetzstein, G., Lin, T.Y.: Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. arXiv preprint arXiv:2503.22020 (2025)
64. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigpt-4: Enhancing vision-language understanding with advanced large language models. arXiv preprint arXiv:2304.10592 (2023)