NoRA: Learning Noise-Resilient Representation for Real-World Depth Perception

Anonymous authors
Paper under double-blind review

Abstract

Simulation enables scalable robot learning, but policies trained on pristine depth can fail under sensor-dependent noisy depth corruption in the real world. Restoring each input adds a denoising stage before control. Inspired by humans achieving robust perception through stable visual cues rather than reconstructing every pixel, we exploit a fundamental physical fact: sensing corruption changes measurements without changing scene geometry. We introduce **NoRA** (**No**ise-Resilient **R**epresentation **A**lignment), which learns stable geometric representations for control without external depth restoration. Its *Noise-Manifold Projection* combines teacher-guided spatial weighting and low-rank projection to suppress sensing-sensitive variation while retaining stable teacher content. NoRA distills clean geometric knowledge, aligns paired clean-reference and real-noisy depth at the policy input, and freezes the complete operator for simulation-only policy learning and subsequent zero-shot real-world transfer. NoRA provides eight CNN and Transformer backbones for different deployment budgets. Evaluations on depth perception, robot manipulation and multi-camera deployment show improved noisy-depth transfer. Under the evaluated sensing perturbations, selective alignment improves noisy geometry and real control over full-feature alignment, at a modest cost in clean geometry accuracy, revealing a practical geometry–robustness trade-off.

1 Introduction

Depth provides geometry for robotic interaction (Patel et al., 2026), while GPU-accelerated simulation enables scalable policy learning (Makoviychuk et al., 2021; Mittal et al., 2025). Transferring these policies to physical robots, however, exposes them to missing measurements and depth distortions (Dai et al., 2022; Liu et al., 2026). Human perception offers an intuitive example: when reaching for a cup under glare, a person can rely on its visible rim and outline without resolving every obscured surface detail. The underlying physical fact is that sensing corruption changes measurements without changing scene geometry. For corresponding observations, geometric evidence therefore need not be equally reliable. Human vision exploits shape cues for recognition (Cutler et al., 2026), and visually guided action depends on the reliability of available evidence (Akande et al., 2025). Fig. 1 illustrates this motivation: preserving useful geometry while reducing dependence on sensing-sensitive evidence allows robots to act on imperfect depth.

Existing approaches address this gap through observation synthesis, restoration, and representation learning. Domain randomization and depth synthesis augment policy training without deployment-time generators (Tobin et al., 2017; Shen et al., 2022; Dai et al., 2022). Restoration methods instead repair real depth before downstream inference: SwinDRNet and CDMs (Dai et al., 2022; Liu et al., 2026), for example, introduce an additional denoising network ahead of the perception and control pipeline. This extra denoising reconstructs depth at every inference step, adding computation and another model to deploy. Depth pretraining offers a complementary route, with DeFM (Patel et al., 2026) learning transferable geometric features, while domain adaptation reduces feature discrepancies across domains (Ganin et al., 2016; Sun & Saenko, 2016). These advances motivate a more selective alignment objective: spatial locations and feature directions need not be equally reliable, and matching intermediate features need not align the final input read by a policy. This motivates our work:

How can we preserve geometry and align reliable evidence directly at the policy's perceptual interface, without an additional depth-restoration stage?

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.

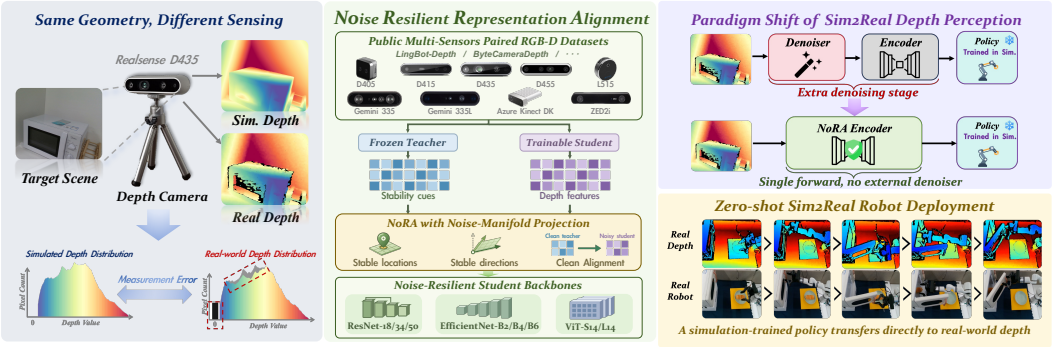

Figure 1: **From Depth Restoration to Noise-Resilient Representations.** Sensor corruption changes depth measurements, while scene geometry remains unchanged. NoRA selects stable locations and feature directions for zero-shot sim2real policy transfer without an external denoiser.

To answer this question, we introduce **NoRA** (**No**ise-Resilient **R**epresentation **A**lignment), a three-stage framework for depth-based policy sim-to-real transfer. Fig. 2 illustrates the framework. ***Stage 1*** distills a frozen DeFM-L/14 teacher into a student on clean simulated depth, establishing a geometric prior. ***Stage 2*** uses paired clean-reference and real-noisy observations to teach the backbone to produce stable geometric features from noisy depth inputs. ***Stage 3*** freezes the complete feature operator and trains the policy on simulated trajectories, preserving alignment during control learning. After encoder alignment, the policy transfers zero-shot to real depth, using the same frozen operator without external restoration for denoising or real-robot policy training.

At NoRA's core, *Noise-Manifold Projection* (NMP) combines *where to trust* with *which feature directions to retain*. Paired teacher differences supervise a spatial gate and a shared low-rank orthogonal basis, selecting stable locations and suppressing sensing-sensitive directions. Context filling preserves the spatial grid. **Noise-Inclusive Alignment (NIA)** retains full spatial support and an identity basis; **Noise-Exclusive Alignment (NEA)** learns the gate and basis. Clean-teacher supervision guides noisy student features, with optional reciprocal matching; a clean-view anchor regularizes geometric knowledge, while normalized consistency constrains the full policy interface. At deployment, reliability is predicted from a single observation with a fixed basis.

NoRA provides a backbone family spanning **ResNet-18/34/50**, **EfficientNet-B2/B4/B6**, and **ViT-S/14 and L/14** for different deployment budgets. Evaluations cover frozen classification and segmentation, spatial completion and temporal feature prediction, sim-to-real manipulation, multi-camera transfer, and deployment efficiency. Ablations examine selection, supervision, and interface preservation. The NEA–NIA comparison identifies a geometry–robustness trade-off: under the evaluated sensing perturbations, selective alignment improves noisy-input geometry and real control while moderately increasing clean-input geometry error relative to full-feature alignment.

Our contributions are threefold:

1) Restoration-free Noise-Resilient Framework. Building on DeFM, NoRA aligns the geometric interface shared by simulation-trained policies and real-depth deployment without an additional denoising network. Its three-stage design establishes geometric knowledge, aligns sensing conditions, and preserves the complete interface during policy learning, enabling sim-to-real transfer without real-robot policy training or teacher inference at deployment.

2) Stability-guided Noise-Manifold Projection. NMP selects teacher-stable evidence across spatial locations and feature directions using paired agreement. Clean-teacher supervision guides noisy student features, while clean-view anchoring regularizes geometric content and normalized consistency constrains the interface the policy reads. This connects noise suppression to a geometrically informative, deployment-ready representation while retaining the dense spatial interface.

3) Improved Robust Perception and Sim2Real Transfer. We provide eight CNN and Transformer backbones and demonstrate improved depth perception and sim-to-real control, supported by efficiency, cross-sensor, and ablation studies. The NEA–NIA comparison quantifies a trade-off: improved noisy geometry and real control accompany a moderate increase in clean geometry error under the evaluated sensing perturbations.

2 Related Work

2.1 Sim2Real Transfer for Depth Perception

Depth-based policy transfer is challenged by sensor-dependent geometric errors and missing measurements. According to whether synthetic or real observations are modified, observation-level approaches follow two complementary routes. ***Depth simulation*** makes training observations more realistic. DepthSynth (Planche et al., 2017), active stereo simulation (Zhang et al., 2023), and DREDS (Dai et al., 2022) model image formation and material-dependent artifacts. Beyond explicit simulation, DCL (Shen et al., 2022) learns geometry-preserving depth synthesis from unpaired data, while Stable-Sim2Real (Xu et al., 2025) refines depth residuals through diffusion and DiffuDepGrasp (Zhou et al., 2025) grafts learned noise onto clean geometry. PRISM (Liang et al., 2026) separates sensing invalidation from measurement inaccuracy, using semantic-geometric priors to guide synthesis. Training-time synthesis adds no deployment generator; DeFM also augments policy depth with noise (Patel et al., 2026). ***Depth restoration*** instead denoises real observations before downstream processing. SwinDR-Net (Dai et al., 2022) restores depth from RGB-D inputs; R2SGrasp (Cai et al., 2025) couples repair with simulated feature memory for grasping; and CDMs (Liu et al., 2026) enable simulation-trained manipulation through RGB-D metric-depth correction. PromptDA and PriorDA (Lin et al., 2025b; Wang et al., 2025) exploit foundation depth priors, while LingBot-Depth (Tan et al., 2026) learns completion and multimodal representations. NoRA complements these observation-level routes by aligning depth features before simulation-only policy learning, without requiring an external restorer at deployment.

2.2 Representation Alignment for Sim2Real Transfer

Representation learning seeks to preserve useful scene structure while reducing sensitivity to domain variation. Based on their learning objectives, related approaches form three overlapping groups. ***Domain adaptation*** aligns source and target features through adversarial confusion in DANN (Ganin et al., 2016), correlation matching in Deep CORAL (Sun & Saenko, 2016), and subspace mapping (Fernando et al., 2013). GL-MSDA (Ma et al., 2024) extends alignment to global scene and local grasp features. ***State representation learning*** constructs compact inputs for control. Jang et al. (2024) combine translated depth with latent alignment during navigation learning, while Deep Collision Encoding (Kulkarni & Alexis, 2024) uses robot-size-aware collision supervision. Their latent spaces thus reflect behavioral or collision objectives. ***Information gating*** identifies which evidence to retain: InfoGating (Tomar et al., 2023) restricts inputs or activations using dynamics and control objectives, MaDi (Grooten et al., 2024) learns reward-driven masks, SGQN (Bertoin et al., 2022) uses decision saliency, and DeGuV (Pham et al., 2025) learns depth-guided masks for visual reinforcement learning. NoRA uses paired teacher agreement to supervise spatial reliability and a shared stable subspace before policy learning. This supervision requires paired depth, but no reward, action, or collision labels. Unlike mapping separate domain subspaces, it learns one orthogonal basis that suppresses paired sensing differences while preserving shared geometric content. Soft gating preserves the token grid, and the operator stays frozen during policy learning; stability remains a proxy requiring downstream validation.

2.3 Visual and Depth Pretraining for Robot Learning

Pretraining supplies reusable perceptual structure for robot learning, reducing the burden on downstream policy optimization. According to input modalities and data representations, this literature comprises four related groups. ***RGB pretraining*** produces transferable visual features through R3M (Nair et al., 2023), VC-1 (Majumdar et al., 2023), and multi-teacher distillation in Theia (Shang et al., 2025). ***3D pretraining*** operates on spatial representations: DepthContrast (Zhang et al., 2021) learns point-cloud and voxel features, while SUGAR (Chen et al., 2024) combines geometry, semantics, and affordances. ***Multimodal pretraining*** integrates complementary inputs: MultiMAE (Bachmann et al., 2022) uses masked cross-modal prediction, and Vanishing Depth (Koch & Krüger, 2026) equips pretrained RGB encoders with metric-depth adapters. ***Depth pretraining*** learns directly from depth images: DeFM (Patel et al., 2026) provides DINO-style features, compact students, and cross-sensor generalization. These input modalities yield distinct geometric interfaces for downstream policies. Building on DeFM's existing prior, NoRA adds paired stability alignment and retains the resulting feature operator for policy training and deployment.

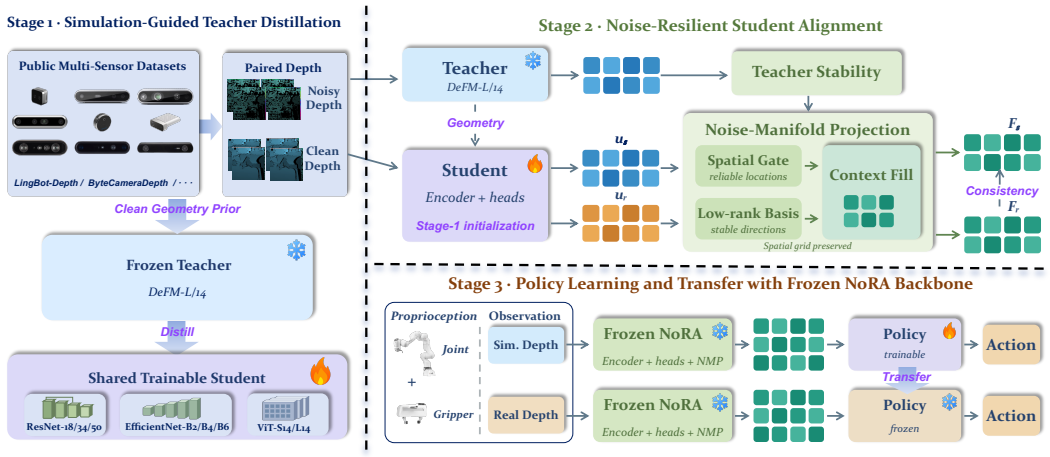

Figure 2: **Overview of NoRA.** Stage 1 distills DeFM on rendered depth. The dataset inset supplies Stage-2 pairs from LingBot-Depth and ByteCameraDepth. Stage 2 learns spatial gates and a low-rank basis through paired alignment, geometry anchoring, and normalized consistency. Stage 3 freezes perception for simulation-trained policy transfer without external restoration.

3 Methodology

3.1 Problem Setup

Problem Setting. NoRA treats depth Sim2Real transfer as a representation-interface problem rather than an image-restoration problem. Fig. 2 illustrates the framework. Our objective is to learn a feature operator whose output preserves task-relevant geometry while remaining insensitive to sensor-dependent corruption. Let $\mathcal{D}_{\text{sim}}$ contain clean simulated depth and $\mathcal{D}_{\text{align}} = \{(\mathbf{X}_s, \mathbf{X}_r)\}$ contain spatially corresponding clean-reference and real-noisy observations of the same scene. Our operator F_Ω makes these inputs compatible for simulation-trained policies. Pairs supervise representation learning; deployment uses a single observation. Token-wise supervision requires spatial correspondence, since misregistration can contaminate the stability cue. App. D.1.1 details pair construction.

Feature Representation. The frozen DeFM-L/14 (Patel et al., 2026) teacher T supplies geometric supervision, whereas the shared student S_θ is the deployable encoder. Both process log-compressed depth $\psi(\mathbf{X}_v)$ for view $v \in \{s, r\}$. The teacher returns a global feature z_v^g and N patch tokens $z_{v,i}^p$; the student returns h_v^g and $h_{v,i}^p$. To bridge feature dimensions, learned heads give $u_v^g = P_{\text{global}}(h_v^g)$ and $u_{v,i}^p = P_{\text{patch}}(h_{v,i}^p) \in \mathbb{R}^{D_t}$. We write $\bar{a} = a/\|a\|_2$.

3.2 Stage 1: Simulation-Guided Teacher Distillation

Before learning sensor invariance, we first teach the student the geometric structure available in clean simulation. Separating these objectives gives Stage 2 a geometric reference to preserve: otherwise it must learn scene structure and suppress sensing variation simultaneously, obscuring whether a discrepancy reflects missing geometry or sensor corruption.

We distill the frozen teacher on $\mathbf{X}_s \sim \mathcal{D}_{\text{sim}}$ using complementary supervision following the DeFM crop recipe. $\mathcal{L}_{\text{Global}}$ transfers scene-level structure across global crops. $\mathcal{L}_{\text{Local}}$ matches partial observations to global teacher features, encouraging local-to-global consistency. $\mathcal{L}_{\text{Patch}}$ preserves spatial geometry by matching corresponding dense tokens. Finally, $\mathcal{L}_{\text{KoLeo}}$ encourages feature diversity and discourages collapse. Together, these terms establish the clean geometric prior:

$$\mathcal{L}_{\text{Distill}} = \mathcal{L}_{\text{Global}} + \mathcal{L}_{\text{Local}} + \mathcal{L}_{\text{Patch}} + \lambda_k \mathcal{L}_{\text{KoLeo}}. \tag{1}$$

App. C.1 defines all four terms. We carry the EMA student checkpoint into Stage 2 and use the same checkpoint as the Pure-Sim baseline. This controls clean pretraining initialization; matched-budget ablations distinguish extra optimization, paired sensing exposure, and selective alignment.

3.3 Stage 2: Noise-Resilient Alignment with NMP

Stage 2 trains the backbone to express clean geometry from real noisy depth, retaining the Stage-1 prior. Sensing corruption affects both spatial locations and feature directions. NMP therefore combines a *where to trust* gate with a shared low-rank *which directions to trust* subspace. We develop the selective operator below and express both NEA and its full-support reference, NIA.

Cross-View Stability as Spatial Reliability. For corresponding observations, frozen-teacher disagreement reflects paired sensing and reference variation. Teacher–teacher residuals separate this stability cue from the student's changing fitting error. We define

$$d_i = \|\bar{z}^p_{s,i} - \bar{z}^p_{r,i}\|_2, \qquad s_i = \mathrm{stopgrad}[\exp(-d_i/\tau_s)]\,. \tag{2}$$

Small d_i indicates teacher-relative stability. Let $y^\rho_i \in \{0,1\}$ mark the top $\lfloor \rho N \rfloor$ stability scores. A shared gate $m_{v,i} = G_\phi(h^p_{v,i}) \in (0,1)$ learns this paired target from each view independently, enabling deployment without a teacher or clean reference. Each $m_{v,i}$ controls its own branch's output; $w_i = (m_{s,i} + m_{r,i})/2$ weights paired alignment. App. C.2.1 details target construction and role of τ_s.

Four constraints guide the gate: stability supervision favors stable locations; cross-view agreement encourages compatible predictions; a retention budget discourages all-on/all-off solutions; and entropy minimization encourages decisive weights:

$$\begin{aligned}\mathcal{R}_{\mathrm{gate}} = \frac{1}{N}\sum_i [\mathrm{BCE}(m_{s,i}, y^\rho_i) + \mathrm{BCE}(m_{r,i}, y^\rho_i)] + \frac{\lambda_{gc}}{N}\sum_i (m_{s,i} - m_{r,i})^2 \\ + \lambda_b(\bar{m} - \rho)^2 + \frac{\lambda_e}{N}\sum_i H_{\mathrm{Bern}}(w_i), \quad \bar{m} = N^{-1}\sum_i w_i.\end{aligned} \tag{3}$$

Here BCE denotes binary cross-entropy and H_{Bern} Bernoulli entropy; App. C.2.1 details their definitions. The budget controls average retained mass, whereas the entropy term discourages ambiguous gate values. Both are soft constraints: the learned weights need not retain exactly ρN tokens, and no tokens are physically removed.

Stable Feature-Subspace Projection. Spatial stability does not imply that every feature direction is reliable. NMP seeks a subspace that rejects sensing-dependent displacement while retaining shared content at stable locations. We learn $\Phi \in \mathbb{R}^{D_t \times r}$ with $\Phi^\top\Phi = I_r$ and $Q_\Phi = \Phi\Phi^\top$. For $\Delta^t_i = \bar{z}^p_{r,i} - \bar{z}^p_{s,i}$ and $\mu^t_i = (\bar{z}^p_{r,i} + \bar{z}^p_{s,i})/2$, these two requirements give

$$\mathcal{R}_{\mathrm{proj}} = \frac{1}{N}\sum_i \left[(1 - s_i)\|Q_\Phi \Delta^t_i\|_2^2 + \lambda_{\mathrm{keep}} s_i \|(I - Q_\Phi)\mu^t_i\|_2^2\right]. \tag{4}$$

The $(1 - s_i)$ term rejects sensing-sensitive displacement, while the s_i term preserves stable content: suppressing displacement alone could retain an uninformative subspace. Orthonormality fixes the basis scale, and rank $r < D_t$ controls the capacity–stability trade-off. The learned linear basis is shared across inputs and fixed at deployment; App. C.2.2 details optimization.

Constructing the Deployable Feature Operator. Hard token removal would disrupt the spatial interface used by dense heads and policies. Instead, we attenuate unreliable local evidence while preserving the grid. Define $a^g_v = \Phi^\top u^g_v$, $a_{v,i} = \Phi^\top u^p_{v,i}$, and the all-token context $c_v = N^{-1}\sum_i a_{v,i}$. The actual downstream representation is

$$F_{\theta,\phi,\Phi}(\mathbf{X}_v) = \mathcal{A}\big(a^g_v, \{m_{v,i}a_{v,i} + (1 - m_{v,i})c_v\}_{i=1}^N\big)\,. \tag{5}$$

Context filling mixes local and scene features while preserving spatial indexing and tensor dimensions. The same task-specific interface $\mathcal{A}$ is used for alignment, policy learning, and deployment: F is the actual downstream representation. App. C.2.2 details interpolation and coordinate handling.

Alignment with Geometry and Interface Preservation. Three constraints favor noisy-input features geometrically informative and stable. ***First***, bidirectional matching prioritizes noisy-student supervision from clean teacher geometry; the other direction transfers retained noisy-teacher evidence to the clean student. With $\hat{u} = Q_\Phi u$ and $\hat{z} = Q_\Phi z$, we match global and local features using $\boldsymbol{\ell_{\mathrm{match}}}$, combining cosine distance and SmoothL1. The mixing weight $\alpha = 2/3$ gives noisy-to-clean supervision twice the emphasis of its reverse. App. C.2.3 details feature normalization, loss reductions, and gradient paths through the student, spatial gate, and shared projection basis.

$$\mathcal{L}_{\text{bi-scene}} = \alpha \ell_{\text{match}}(\hat{u}_r^g, \hat{z}_s^g) + (1-\alpha)\ell_{\text{match}}(\hat{u}_s^g, \hat{z}_r^g), \tag{6}$$

$$\mathcal{L}_{\text{bi-token}} = \frac{\sum_i w_i [\alpha \ell_{\text{match}}(\hat{u}_{r,i}^p, \hat{z}_{s,i}^p) + (1-\alpha)\ell_{\text{match}}(\hat{u}_{s,i}^p, \hat{z}_{r,i}^p)]}{\sum_i w_i + \epsilon}. \tag{7}$$

Second, alignment alone can discard useful geometry. An unprojected clean-view anchor regularizes Stage-1 knowledge in the clean student branch, supplying the shared backbone with feature-direction targets before gating or projection:

$$\mathcal{L}_{\text{semantic}} = \ell_{\text{match}}(u_s^g, z_s^g) + \frac{1}{N}\sum_i \ell_{\text{match}}(u_{s,i}^p, z_{s,i}^p). \tag{8}$$

The anchor regularizes the full clean-view representation, while noisy-to-clean matching guides geometry in the real-input branch. Separately, $\mathcal{R}_{\text{proj}}$ preserves stable content within the retained basis. Together they couple noise suppression with geometric information retention.

Third, token matching need not align the aggregated policy interface: gates and context filling also affect its features. A parameter-free loss head ν_ϵ normalizes each of its B_F spatial or pooled blocks $F_{v,b}$; interface consistency compares their directions:

$$\mathcal{L}_{\text{deploy}} = B_F^{-1} \sum\nolimits_{b=1}^{B_F} \left[1 - \langle \nu_\epsilon(F_{s,b}), \nu_\epsilon(F_{r,b}) \rangle\right]. \tag{9}$$

Above the norm floor, feature rescaling cannot lower this directional loss; App. B.1.3 defines its normalization, gradients, and numerical safeguards.

Full-Support versus Selective Alignment. Both variants share a common alignment objective, evaluated with their respective spatial weights m and feature basis Φ:

$$\mathcal{L}_{\text{align}}(m, \Phi) = \lambda_{\text{scene}}\mathcal{L}_{\text{bi-scene}} + \lambda_{\text{token}}\mathcal{L}_{\text{bi-token}} + \lambda_{\text{sem}}\mathcal{L}_{\text{semantic}} + \lambda_{\text{dep}}\mathcal{L}_{\text{deploy}}. \tag{10}$$

NIA uses full spatial support and the identity basis; NEA learns the gate and rank-r basis with the two operator regularizers. The Stage-2 objective $\boldsymbol{\mathcal{L}_{\text{NMP}}^{k}}$ for $k \in \{\text{NIA}, \text{NEA}\}$ is:

$$\mathcal{L}_{\text{NMP}}^{\text{NIA}} = \mathcal{L}_{\text{align}}(\mathbf{1}, I_{D_t}), \quad \mathcal{L}_{\text{NMP}}^{\text{NEA}} = \mathcal{L}_{\text{align}}(m, \Phi) + \lambda_{\text{gate}}\mathcal{R}_{\text{gate}} + \lambda_{\text{proj}}\mathcal{R}_{\text{proj}}. \tag{11}$$

Here m collects the view-specific gates, which also determine w_i. Both variants start from the same Stage-1 checkpoint: NIA aligns all available features, whereas NEA selects stable spatial evidence and feature directions. App. C.2.4–C.2.5 details optimization.

3.4 Stage 3: Policy Learning with a Frozen Feature Interface

Stage 2 establishes the perception interface on which transfer relies. Simulation-only policy fitting could undo this alignment if allowed to change the encoder, gate, or basis. We therefore freeze the EMA student, alignment heads, and selected operator as F_Ω before learning control. Given simulated rollouts $\mathcal{D}_\pi^{\text{sim}} = \{(\mathbf{X}_{s,t}, q_t, a_t)\}$, with proprioception q_t and action target a_t, only policy parameters ω are optimized. Here $\boldsymbol{\mathcal{L}_{\text{Policy}}}$ denotes the action-supervision objective of the chosen policy learner:

$$\min_\omega \mathbb{E}_{\mathcal{D}_\pi^{\text{sim}}} \left[\mathcal{L}_{\text{Policy}}\big(\pi_\omega([F_\Omega(\mathbf{X}_{s,t}), q_t]), a_t\big)\right]. \tag{12}$$

Freezing the entire operator preserves the feature mapping established in Stage 2 while the policy learns task-dependent action prediction. Freezing the backbone alone would leave the interface free to drift through changes to its heads, gate, or basis. This action-supervised policy uses the same feature dimensions and aggregation at training and deployment. Deployment replaces $\mathbf{X}_{s,t}$ with raw real depth $\mathbf{X}_{r,t}$ while keeping the feature operator and policy fixed. It requires neither teacher inference, paired references, nor an external restoration network. The supported setting is *zero-shot policy transfer after real-depth encoder alignment*, with no real-robot policy training or online adaptation.

3.5 NoRA Backbone Family

NoRA is architecture-agnostic with respect to the deployment student, providing a family of frozen depth encoders for different deployment budgets: **ResNet** (18/34/50), **EfficientNet** (B2/B4/B6), and **ViT** (S/14 and L/14). Each backbone follows the same three-stage pipeline with a frozen DeFM-L/14 teacher and supports both NIA and NEA. The resulting deployment model comprises the student backbone, its alignment heads, and the selected feature operator; the teacher is needed only for training. Users select a backbone according to their accuracy–latency budget and train task-specific heads on its frozen representations. App. B.1 details the architecture and feature dimensions.

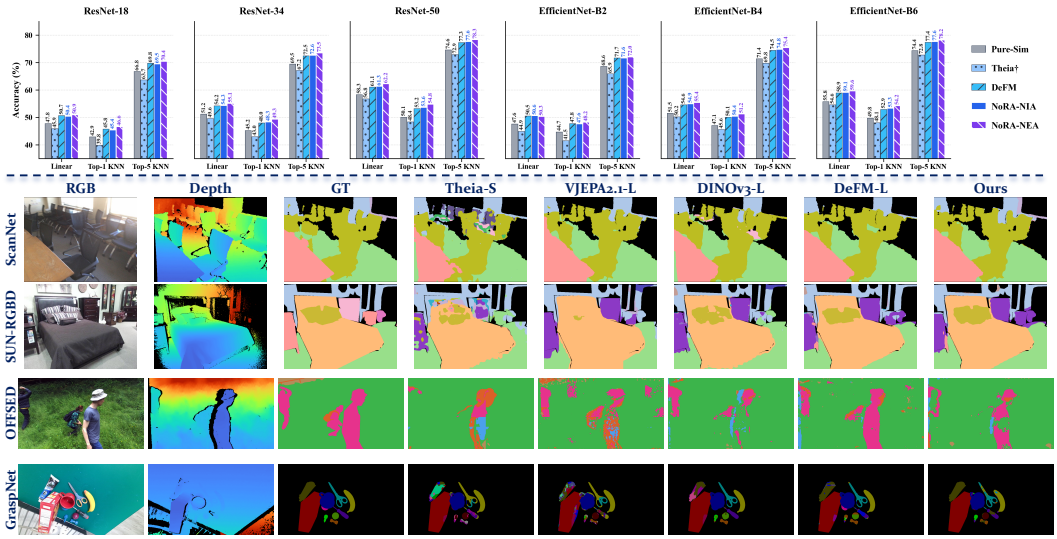

Figure 3: **Visual Transfer of NoRA Backbones.** CNN classification and ViT segmentation.

Table 1: **Frozen-Backbone Depth Segmentation on ViTs across 4 Datasets with 7 Sensor Types.**

Backbone	Method	Indoor		Outdoor	Manipulation	Avg.
		ScanNet	SUN-RGBD	OFFSED	GraspNet-1B	
ViT-S/16	Theia	23.6	22.8	48.7	18.2	28.3
ViT-S/16	DINOv3	28.4	28.9	56.8	22.9	34.3
ViT-S/14	DeFM	29.4	29.7	**57.6**	23.1	35.0
ViT-S/14	**NoRA**	**31.8**	**32.2**	57.1	**26.7**	**37.0**
ViT-L/16	V-JEPA 2.1	32.4	33.2	55.2	28.8	37.4
ViT-L/16	DINOv3	33.2	34.6	55.8	28.3	38.0
ViT-L/14	DeFM	33.6	33.1	58.0	30.6	38.8
ViT-L/14	**NoRA**	**36.0**	**35.8**	**58.5**	**34.1**	**41.1**

4 Experiments

We organize our evaluation around four questions: ***Q1:*** *Does NoRA provide transferable representations for visual understanding?* (Sec. 4.2) ***Q2:*** *Does NoRA support spatial feature completion and temporal forecasting in a shared target space?* (Sec. 4.3) ***Q3:*** *Do NoRA's representations enable robust and efficient real-world control across sensing conditions?* (Sec. 4.4) ***Q4:*** *Which design choices explain NoRA's geometry–robustness trade-off and control performance?* (Sec. 4.5)

4.1 Experimental Settings

We compare ***Pretrained Representation Models***—DeFM (Patel et al., 2026), Theia (Shang et al., 2025), DINOv3 (Siméoni et al., 2026), and V-JEPA 2.1 (Mur-Labadia et al., 2026); and ***Depth Restoration Methods***—CDM (Liu et al., 2026), PromptDA (Lin et al., 2025b), PriorDA (Wang et al., 2025), and LingBot-MDM (Tan et al., 2026). All baselines and NoRA share 30,000 adaptation updates on LingBot-Depth and ByteCameraDepth (Liu et al., 2026), followed by task-specific readouts on frozen feature operators. Robot policies share DP3 (Ze et al., 2024) and clean simulation training. App. D specifies data roles, adaptation, scoring, and the revision scope of the numerical comparisons. NoRA denotes NEA in the main comparisons; NIA is evaluated in the ablations.

4.2 Frozen-Backbone Transfer for Visual Understanding

Image Classification on CNNs. The CNN classification panel compares Pure-Sim, DeFM, Theia, NIA and NoRA on ResNet-18/34/50 (He et al., 2016) and EfficientNet-B2/B4/B6 (Tan & Le, 2019). Following DeFM (Patel et al., 2026), fixed Depth Anything V2 (Yang et al., 2024) converts ImageNet-1K (Russakovsky et al., 2015) into depth for frozen-encoder evaluation. Fig. 3(a) illustrates classification performance; App. D.2.1 details linear Top-1 and kNN Top-1/Top-5 evaluation.

Semantic Segmentation on ViTs. We compare adapted ViT-S/L baselines with NoRA using frozen linear probes. Following DeFM (Patel et al., 2026), evaluation covers ScanNet's Structure Sensor (Dai et al., 2017), SUN-RGBD's Kinect v1/v2, RealSense, and Xtion (Song et al., 2015), OFFSED's ZED stereo (Neigel et al., 2021), and GraspNet-1B's Azure Kinect (Fang et al., 2020). Tab. 1 reports mIoU; Fig. 3(b) illustrates segmentation quality. App. D.2.2 details sensor coverage and probe training.

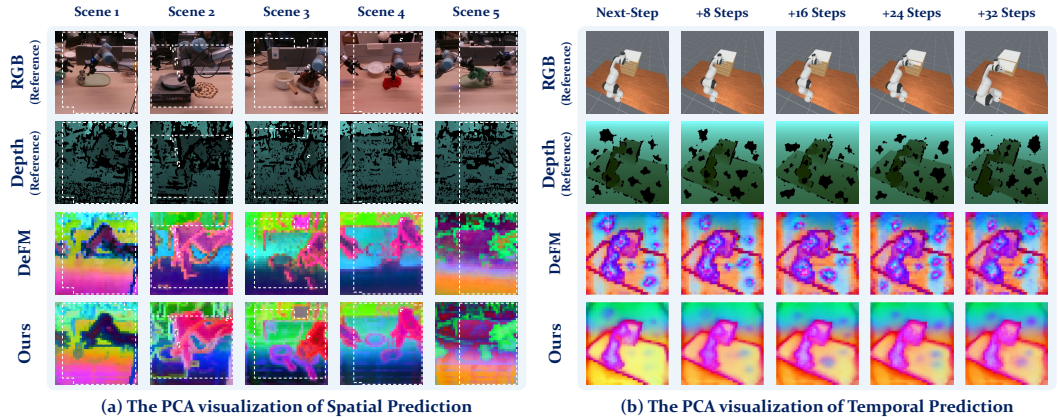

Figure 4: **Qualitative Results of Spatial Completion and Multi-Step Temporal Prediction.** PCA of (a) completion from masked depth observations; (b) multi-step visual forecasts on robot trajectories.

Table 2: **Spatial Completion and Temporal Prediction.** Shared-target *cosine similarity* (↑).

Encoder	Spatial: Mask Ratio			Temporal: Future Steps				
	0.25	0.50	0.75	1	8	16	24	32
PromptDA	0.923	0.901	0.870	0.912	0.802	0.684	0.594	0.545
CDM	0.940	0.920	0.891	0.933	0.832	0.720	0.626	0.573
LingBot-MDM	0.958	0.952	**0.943**	0.965	0.898	0.846	0.810	**0.784**
DINOv3	0.951	0.932	0.902	0.943	0.850	0.735	0.646	0.595
V-JEPA 2.1	0.948	0.931	0.906	0.941	0.852	0.743	0.658	0.602
DeFM	**0.963**	0.947	0.921	0.953	0.880	0.817	0.774	0.741
NoRA	0.960	**0.952**	0.935	**0.967**	**0.903**	**0.849**	**0.812**	0.778

4.3 Representation Quality for Spatio-temporal Prediction

Masked Spatial Feature Completion. We probe frozen representations through RGB-conditioned completion in a shared depth-feature space. A V-JEPA 2.1 predictor (Mur-Labadia et al., 2026) completes depth features on held-out LingBot-Depth scenes (Tan et al., 2026), given 25/50/75% randomly masked depth and complete RGB context. Tab. 2 compares eligible ViT-L models under matched training budgets. Fig. 4(a) illustrates DeFM-L and NoRA-L completions; App. D.3.1 details shared DeFM targets, feature interfaces, and the RGB-conditioned evaluation scope.

Multi-Step Temporal Feature Prediction. We probe how depth representations support forecasting shared future reference features. PRISM (Liang et al., 2026) augments simulated depth from ManiSkill3 (Tao et al., 2025) with realistic sensor noise. From eight historical depth frames, each model predicts shared clean-depth reference features at 1/8/16/24/32 future steps, without RGB or future actions. Tab. 2 reports similarity across horizons; Fig. 4(b) illustrates DeFM-L and NoRA-L forecasts. App. D.3.2 details trajectory generation and the scope of feature-space prediction.

4.4 Sim-to-Real Manipulation and Deployment Efficiency

Simulation-to-Real Manipulation. We train DP3 (Ze et al., 2024) on shared clean-depth Isaac Sim (NVIDIA, 2026) demonstrations for Game, Canteen, and Kitchen. Deployment uses a RealMan RM75-6F arm and Pika gripper. Each frozen encoder conditions an independently fitted policy under the same budget. Restoration methods precede DeFM-L feature extraction. NoRA improves on adapted DeFM by 10.0–12.2 points across camera averages (Tab. 3) at similar latency; App. D.4.1 specifies the shared protocol and cumulative task scoring.

Multi-Camera Generalization and Efficiency. Fixed policies transfer across D435, L515, Gemini 336, ZED 2i, and Azure Kinect, with 30 trials per task/camera. Four models appear in adaptation; Gemini 336 is held out within a seen family. NoRA and LingBot-MDM average 80.0% and 78.0% success; NoRA reduces measured latency by 46.8% (Tab. 3). Apps. D.4.2 and D.4.3 specify camera exposure, calibration, and pipeline timing.

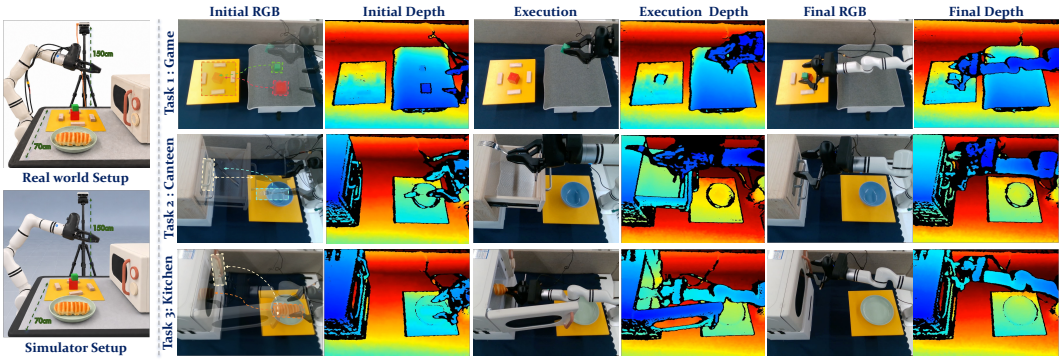

Figure 5: **Sim2Real Setup and RGB-D Rollouts of Manipulation Tasks. Left**: simulation and real-world setups. **Right**: RGB-D observations during real-robot manipulation tasks.

Table 3: **Zero-shot Sim2Real Robotic Manipulation on 3 Tasks across 5 Cameras.** *Success rate* (%), mean±std over 3 seeds with 10 trials. *Lat.* (ms) and *Param* (M) include the pipeline and policy.

Method	Lat.	Params	RealSense D435				Depth Camera Types			
			Game	Canteen	Kitchen	Average	L515	G336	ZED 2i	Kinect
PriorDA-B	175.6	575.97	70.0 ± 10.0	63.3 ± 5.8	53.3 ± 5.8	62.2 ± 1.9	58.9 ± 5.1	62.2 ± 3.9	52.2 ± 5.1	61.1 ± 1.9
PromptDA-L	148.3	721.21	73.3 ± 11.6	56.7 ± 5.8	50.0 ± 10.0	60.0 ± 3.3	61.1 ± 5.1	62.2 ± 1.9	52.2 ± 1.9	61.1 ± 1.9
CDM-L	187.8	1040.53	83.3 ± 5.8	73.3 ± 5.8	73.3 ± 11.6	76.7 ± 3.3	75.6 ± 1.9	73.3 ± 0.0	70.0 ± 3.3	75.6 ± 1.9
LingBot-MDM-L	192.6	702.33	80.0 ± 10.0	**80.0** ± 10.0	76.7 ± 5.8	78.9 ± 1.9	78.9 ± 3.9	80.0 ± 6.7	74.4 ± 3.9	77.8 ± 1.9
Theia-B	97.4	162.19	70.0 ± 10.0	60.0 ± 10.0	53.3 ± 11.6	61.1 ± 8.4	57.8 ± 5.1	61.1 ± 1.9	52.2 ± 5.1	58.9 ± 3.9
DINOv3-L	108.3	380.66	76.7 ± 5.8	56.7 ± 15.3	56.7 ± 5.8	63.3 ± 6.7	64.4 ± 5.1	65.6 ± 1.9	56.7 ± 6.7	64.4 ± 3.9
V-JEPA 2.1-L	170.4	380.62	76.7 ± 5.8	66.7 ± 11.6	63.3 ± 5.8	68.9 ± 3.9	65.6 ± 3.9	68.9 ± 5.1	63.3 ± 3.3	67.8 ± 1.9
DeFM-L	101.7	381.16	80.0 ± 10.0	66.7 ± 15.3	66.7 ± 5.8	71.1 ± 1.9	67.8 ± 8.4	70.0 ± 12.0	65.6 ± 5.1	70.0 ± 13.3
NoRA-L	102.4	381.66	**86.7** ± 5.8	76.7 ± 5.8	**80.0** ± 10.0	**81.1** ± 3.9	**80.0** ± 3.3	**81.1** ± 3.9	**76.7** ± 3.3	**81.1** ± 3.9



Figure 6: **Ablation Results.** Three-seed *means*; CGeo/NGeo are clean/noisy geometry RMSE (mm, ↓), and Real is task-averaged success rate (%). Ex.4 is NIA and Ex.20 is full NEA. Details in App. E.

4.5 Ablation Study

Effect of Paired Alignment and Operator Components. Ex.1–Ex.6 compare extra training, synthetic versus observed pairs, and operator components; NIA is Ex.4 and NEA is Ex.20. Tab. A.9 measures geometry and control under matched readouts. Ex.3 changes representation adaptation while policies train on clean depth; App. E.1 specifies these interventions.

Effect of Stability Supervision and Projection. Ex.7–Ex.14 test whether teacher stability improves spatial selection and preserves useful feature directions beyond ordinary compression. Matched-rank bases, shuffled targets, and separate projection terms test their contributions; Tab. A.10 and App. E.2 report geometry-probe errors and real control success.

Effect of Geometry Anchoring and Policy Interfaces. Ex.15–Ex.19 compare anchoring, direction, interface regularization, and freezing against Ex.20. One-way matching slightly improves on the default; reciprocal supervision remains optional. Tab. A.11 links geometry with simulation and real control; App. E.3 distinguishes these retained comparisons from the revised consistency objective.

5 Conclusion

NoRA improves noisy-depth transfer through stability-guided selective alignment, trading some clean-input geometry accuracy for noise robustness. Its frozen perception interface supports simulation-trained control across the evaluated cameras without external restoration. Limits of reference supervision, sensing coverage, and robot tasks are discussed in App. F.

STATEMENT ON THE USE OF GENERATIVE AI

In this work, we used generative AI tools (Codex) only to assist in drafting parts of the manuscript, edit the text for readability, and adjust LaTeX formatting and layout. We did not use generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims, provide ingredients for mathematical proofs or assist in writing proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement the NoRA method, or interpret results. We did not use generative AI tools to create or modify figures or plotting code. All AI-assisted manuscript preparation was conducted under human direction and supervision. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Abibat A. Akande, Philipp Kreyenmeier, and Miriam Spering. Visual factors that determine sensory uncertainty in rapid interceptive hand movements. *Journal of Vision*, 25(4):8, 2025. doi: 10.1167/jov.25.4.8. URL https://doi.org/10.1167/jov.25.4.8.

Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. MultiMAE: Multi-modal multi-task masked autoencoders. In *European Conference on Computer Vision (ECCV)*, 2022. URL https://arxiv.org/abs/2204.01678.

David Bertoin, Adil Zouitine, Mehdi Zouitine, and Emmanuel Rachelson. Look where you look! saliency-guided Q-networks for generalization in visual reinforcement learning. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2209.09203.

Jia-Feng Cai, Zibo Chen, Xiao-Ming Wu, Jian-Jian Jiang, Yi-Lin Wei, and Wei-Shi Zheng. Real-to-sim grasp: Rethinking the gap between simulation and real world in grasp detection. In *Proceedings of The 8th Conference on Robot Learning*, volume 270, pp. 1109–1124. PMLR, 2025. URL https://proceedings.mlr.press/v270/cai25a.html.

Shizhe Chen, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. SUGAR: Pre-training 3D visual representations for robotics. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. URL https://arxiv.org/abs/2404.01491.

Mikayla Cutler, Luke Baumel, Joseph Tocco, William Friebel, George K. Thiruvathukal, and Nicholas Baker. Three-dimensional shape cues affect human and artificial recognition systems differently. *PLOS ONE*, 21(5):e0338885, 2026. doi: 10.1371/journal.pone.0338885. URL https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0338885.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Niessner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5828–5839, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Dai_ScanNet_Richly-Annotated_3D_CVPR_2017_paper.html.

Qiyu Dai, Jiyao Zhang, Qiwei Li, Tianhao Wu, Hao Dong, Ziyuan Liu, Ping Tan, and He Wang. Domain randomization-enhanced depth simulation and restoration for perceiving and grasping specular and transparent objects. In *European Conference on Computer Vision (ECCV)*, 2022. URL https://arxiv.org/abs/2208.03792.

Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. GraspNet-1Billion: A large-scale benchmark for general object grasping. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11444–11453, 2020. URL https://graspnet.net/.

Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 2960–2967, 2013. URL https://openaccess.thecvf.com/content_iccv_2013/html/Fernando_Unsupervised_Visual_Domain_2013_ICCV_paper.html.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

Bram Grooten, Tristan Tomilin, Gautham Vasan, Matthew E. Taylor, A. Rupam Mahmood, Meng Fang, Mykola Pechenizkiy, and Decebal Constantin Mocanu. MaDi: Learning to mask distractions for generalization in visual deep reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2024. URL https://arxiv.org/abs/2312.15339.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Yoonsu Jang, Jongchan Baek, Soo Jeon, and Soohee Han. Bridging the simulation-to-real gap of depth images for deep reinforcement learning. *Expert Systems with Applications*, 253:124310, 2024. doi: 10.1016/j.eswa.2024.124310.

Paul Koch and Jörg Krüger. Vanishing depth: Training generalized depth adapters with sinusoidal depth preprocessing for pretrained RGB encoders. *arXiv preprint arXiv:2503.19947*, 2026. URL https://arxiv.org/abs/2503.19947v2. Version 2; accepted to IntelliSys 2026.

Mihir Kulkarni and Kostas Alexis. Reinforcement learning for collision-free flight exploiting deep collision encoding. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024. URL https://arxiv.org/abs/2402.03947.

Xiujian Liang, Jiacheng Liu, Mingyang Sun, Qichen He, Anda Cheng, Cewu Lu, and Jianhua Sun. PRISM: Learning realistic depth via physics-grounded noise disentanglement with semantic-geometric collaboration. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=AnofTirXgv.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025a.

Haotong Lin, Sida Peng, Jingxiao Chen, Songyou Peng, Jiaming Sun, Minghuan Liu, Hujun Bao, Jiashi Feng, Xiaowei Zhou, and Bingyi Kang. Prompting depth anything for 4K resolution accurate metric depth estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17070–17080, 2025b. URL https://openaccess.thecvf.com/content/CVPR2025/html/Lin_Prompting_Depth_Anything_for_4K_Resolution_Accurate_Metric_Depth_Estimation_CVPR_2025_paper.html.

Minghuan Liu, Zhengbang Zhu, Xiaoshen Han, Peng Hu, Haotong Lin, Xinyao Li, Jingxiao Chen, Jiafeng Xu, Yichu Yang, Yunfeng Lin, Xinghang Li, Yong Yu, Weinan Zhang, Tao Kong, and Bingyi Kang. Manipulation as in simulation: Enabling accurate geometry perception in robots. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=sWyX1BpeN4.

Haoxiang Ma, Ran Qin, Modi Shi, Boyang Gao, and Di Huang. Sim-to-real grasp detection with global-to-local RGB-D adaptation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024. URL https://arxiv.org/abs/2403.11511.

Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Yecheng Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Tingfan Wu, Jay Vakil, Pieter Abbeel, Jitendra Malik, Dhruv Batra, Yixin Lin, Oleksandr Maksymets, Aravind Rajeswaran, and Franziska Meier. Where are we in the search for an artificial visual cortex for embodied intelligence? In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/022ca1bed6b574b962c48a2856eb207b-Abstract-Conference.html.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac Gym: High performance GPU based physics simulation for robot learning. In

Joaquin Vanschoren and Serena Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/28dd2c7955ce926456240b2ff0100bde-Paper-round2.pdf.

Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, et al. Isaac Lab: A GPU-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025. URL https://arxiv.org/abs/2511.04831.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026. URL https://arxiv.org/abs/2603.14482.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In *Proceedings of The 6th Conference on Robot Learning*, volume 205, pp. 892–909. PMLR, 2023. URL https://proceedings.mlr.press/v205/nair23a.html.

Peter Neigel, Jason Raphael Rambach, and Didier Stricker. OFFSED: Off-road semantic segmentation dataset. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, pp. 552–557, 2021. URL https://www.dfki.de/en/web/research/projects-and-publications/publication/11376.

NVIDIA. NVIDIA Isaac Sim documentation, 2026. URL https://docs.isaacsim.omniverse.nvidia.com/latest/index.html. Accessed September 25, 2026.

Manthan Patel, Jonas Frey, Mayank Mittal, Fan Yang, Alexander Hansson, Amir Bar, Cesar Cadena, and Marco Hutter. Defm: Learning foundation representations from depth for robotics. *arXiv preprint arXiv:2601.18923*, 2026. URL https://arxiv.org/abs/2601.18923.

Tien Pham, Xinyun Chi, Khang Nguyen, Manfred Huber, and Angelo Cangelosi. DeGuV: Depth-guided visual reinforcement learning for generalization and interpretability in manipulation. *arXiv preprint arXiv:2509.04970*, 2025. URL https://arxiv.org/abs/2509.04970.

Benjamin Planche, Ziyan Wu, Kai Ma, Shanhui Sun, Stefan Kluckner, Oliver Lehmann, Terrence Chen, Andreas Hutter, Sergey Zakharov, Harald Kosch, and Jan Ernst. DepthSynth: Real-time realistic synthetic data generation from CAD models for 2.5D recognition. In *International Conference on 3D Vision (3DV)*, pp. 1–10, 2017. doi: 10.1109/3DV.2017.00011. URL https://doi.org/10.1109/3DV.2017.00011.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y. URL https://www.image-net.org/challenges/LSVRC/.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Spreading vectors for similarity search. In *International Conference on Learning Representations (ICLR)*, 2019.

SAPIEN Team. PartNet-Mobility: Official dataset. Hugging Face dataset repository, 2026. URL https://huggingface.co/datasets/sapien-sim/PartNetMobility. Accessed September 25, 2026.

Jinghuan Shang, Karl Schmeckpeper, Brandon B. May, Maria Vittoria Minniti, Tarik Kelestemur, David Watkins, and Laura Herlant. Theia: Distilling diverse vision foundation models for robot learning. In *Proceedings of The 8th Conference on Robot Learning*, volume 270, pp. 724–748. PMLR, 2025. URL https://proceedings.mlr.press/v270/shang25a.html.

Yuefan Shen, Yanchao Yang, Youyi Zheng, C. Karen Liu, and Leonidas J. Guibas. DCL: Differential contrastive learning for geometry-aware depth synthesis. *IEEE Robotics and Automation Letters*, 7 (2):4845–4852, 2022. doi: 10.1109/LRA.2022.3148788. URL https://arxiv.org/abs/2107.13087.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seung Eun Yi, Michael Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Herve Jegou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=2NlGyqNjns. Featured Certification.

Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 567–576, 2015. URL https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Song_SUN_RGB-D_A_2015_CVPR_paper.pdf.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision Workshops (ECCVW)*, 2016.

Bin Tan, Changjiang Sun, Xiage Qin, Hanat Adai, Zelin Fu, Tianxiang Zhou, Han Zhang, Yinghao Xu, Xing Zhu, Yujun Shen, and Nan Xue. Masked depth modeling for spatial perception. *arXiv preprint arXiv:2601.17895*, 2026. URL https://arxiv.org/abs/2601.17895.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*, 2019.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh Nagaswamy Rajesh, Yong Woo Choi, Yen-Ru Chen, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. ManiSkill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *arXiv preprint arXiv:2410.00425*, 2025. URL https://arxiv.org/abs/2410.00425. Version 2.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

Manan Tomar, Riashat Islam, Matthew E. Taylor, Sergey Levine, and Philip Bachman. Ignorance is bliss: Robust control via information gating. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2303.06121.

Zehan Wang, Siyu Chen, Lihe Yang, Jialei Wang, Ziang Zhang, Hengshuang Zhao, and Zhou Zhao. Depth anything with any prior. *arXiv preprint arXiv:2505.10565*, 2025. URL https://arxiv.org/abs/2505.10565.

Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11097–11107, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Xiang_SAPIEN_A_SimulAted_Part-Based_Interactive_ENvironment_CVPR_2020_paper.html.

Mutian Xu, Chongjie Ye, Haolin Liu, Yushuang Wu, Jiahao Chang, and Xiaoguang Han. StableSim2Real: Exploring simulation of real-captured 3D data with two-stage depth diffusion. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2609–2619, 2025. URL https://mutianxu.github.io/stable-sim2real/.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024. URL https://arxiv.org/abs/2406.09414.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3D Diffusion Policy: Generalizable visuomotor policy learning via simple 3D representations. In *Proceedings of Robotics: Science and Systems*, 2024. URL https://arxiv.org/abs/2403.03954.

Xiaoshuai Zhang, Rui Chen, Ang Li, Fanbo Xiang, Yuzhe Qin, Jiayuan Gu, Zhan Ling, Minghua Liu, Peiyu Zeng, Songfang Han, Zhiao Huang, Tongzhou Mu, Jing Xu, and Hao Su. Close the optical sensing domain gap by physics-grounded active stereo sensor simulation. *IEEE Transactions on Robotics*, 2023. URL https://arxiv.org/abs/2201.11924.

Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3D features on any point-cloud. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. URL https://arxiv.org/abs/2101.02691.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5745–5753, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Zhou_On_the_Continuity_of_Rotation_Representations_in_Neural_Networks_CVPR_2019_paper.html.

Yingting Zhou, Wenbo Cui, Weiheng Liu, Guixing Chen, Haoran Li, and Dongbin Zhao. DiffuDepGrasp: Diffusion-based depth noise modeling empowers Sim2Real robotic grasping. *arXiv preprint arXiv:2511.12912*, 2025. URL https://arxiv.org/abs/2511.12912.

CONTENTS

1 Introduction **1**

2 Related Work **3**
2.1 Sim2Real Transfer for Depth Perception . . . 3
2.2 Representation Alignment for Sim2Real Transfer . . . 3
2.3 Visual and Depth Pretraining for Robot Learning . . . 3

3 Methodology **4**
3.1 Problem Setup . . . 4
3.2 Stage 1: Simulation-Guided Teacher Distillation . . . 4
3.3 Stage 2: Noise-Resilient Alignment with NMP . . . 5
3.4 Stage 3: Policy Learning with a Frozen Feature Interface . . . 6
3.5 NoRA Backbone Family . . . 6

4 Experiments **7**
4.1 Experimental Settings . . . 7
4.2 Frozen-Backbone Transfer for Visual Understanding . . . 7
4.3 Representation Quality for Spatio-temporal Prediction . . . 8
4.4 Sim-to-Real Manipulation and Deployment Efficiency . . . 8
4.5 Ablation Study . . . 9

5 Conclusion **9**

A Notation **16**
A.1 Symbol Notation and Definition . . . 16

B Architecture and Algorithms **16**
B.1 Network Architectures . . . 16
B.2 Algorithms . . . 17

C Training Objectives and Optimization **17**
C.1 Stage 1: Teacher Distillation . . . 17
C.2 Stage 2: Noise-Resilient Alignment . . . 19
C.3 Training the NoRA Feature Operator . . . 20

D Experimental Settings and Evaluation Protocols **20**
D.1 Data, Baselines, and Comparison Scope . . . 20
D.2 Frozen Visual Transfer . . . 22
D.3 Predictive Representation Protocols . . . 24
D.4 Robot Manipulation Protocols . . . 27

E Ablation and Mechanism Analysis **31**
E.1 Paired Alignment and Operator Components . . . 32
E.2 Stability Supervision and Geometry Retention . . . 32
E.3 Geometry Anchoring and the Policy Interface . . . 33

F Limitations **34**

G Extended Robot Results and Visualizations **35**

A NOTATION

A.1 SYMBOL NOTATION AND DEFINITION

Tab. A.1 summarizes the symbols used in the method and experiments.

Table A.1: Summary of Notation and Symbols.

Symbol	Description
$\mathcal{D}_{\text{sim}}$	Clean simulated depth dataset
$\mathcal{D}_{\text{real}}$	Real noisy depth observations
$\mathcal{D}_{\text{align}}$	Paired alignment dataset $\{(\mathbf{X}_s, \mathbf{X}_r)\}$
$\mathbf{X}_s \in \mathbb{R}^{H\times W}$	Clean-reference depth map (simulated in Stage 1)
$\mathbf{X}_r \in \mathbb{R}^{H\times W}$	Real noisy depth map
$\psi(\cdot)$	Depth preprocessing function
T	Frozen DeFM teacher
S_θ	Student encoder with parameters θ
$z_v^g, z_{v,i}^p$	Teacher global/token features for view $v \in \{s, r\}$
$h_v^g, h_{v,i}^p$	Raw student global/token features for view v
$u_v^g, u_{v,i}^p$	Student features projected to the teacher space
d_i, s_i	Cross-teacher residual and derived stability score
y_i^ρ	Top-ρ teacher-stability target
$m_{v,i}, w_i$	View-specific gates and their symmetric alignment weight
G_ϕ	Lightweight gating network
Δ_i^t, μ_i^t	Cross-view teacher displacement and paired mean token
Φ, Q_Φ	Orthonormal stable basis and projector $\Phi\Phi^\top$
$a_v^g, a_{v,i} = \Phi^\top(u_v^g, u_{v,i}^p)$	Low-rank coordinates consumed downstream
$\rho = 0.8$	Target token retention ratio
$F_{\theta,\phi,\Phi}$	Feature operator used by the downstream policy
F_Ω	Frozen NoRA feature operator after Stage 2
π_ω	Policy network with parameters ω

B ARCHITECTURE AND ALGORITHMS

B.1 NETWORK ARCHITECTURES

B.1.1 STUDENT ENCODER CONFIGURATIONS

All NoRA students use 224×224 depth for alignment and default evaluation; Stage-1 local crops use 98×98 inputs. The deployment operator contains the student, alignment heads, and, for NEA, the gate and stable-subspace projector. App. D.4.3 reports complete feature-to-action costs for the evaluated robot pipelines.

B.1.2 PROJECTION HEADS

The specified alignment heads map student features to the teacher dimension ($D_t = 1024$ for DeFM-L/14):

- **Global Projector:** Linear(D_s, D_t) + LayerNorm
- **Patch Projector:** Linear(D_s, D_t) + LayerNorm

The alignment-head LayerNorm uses learnable gain and bias; the consistency head below has no trainable parameters.

B.1.3 Normalized Consistency Head

Write the exported representation as B_F blocks $F_{v,b} \in \mathbb{R}^{C_m}$: one global-descriptor block for classification ($B_F = 1$), one per spatial token for dense/predictive tasks, and the global mean plus sixteen pooled cells for the robot interface ($B_F = 17$). The loss head acts after the gate, context filling, projection, and task-specific aggregation:

$$\nu_\epsilon(x) = \frac{x}{\max(\|x\|_2, \epsilon)}, \qquad \epsilon = 10^{-6}. \tag{13}$$

Norms and dot products are computed in FP32, with gradients through both inputs and normalization. Blocks are weighted uniformly, independently of the gate. Downstream heads consume F; the normalization head is used only in the training loss.

For positive scales c_v with original and scaled norms above ϵ, $\nu_\epsilon(c_v F_{v,b}) = \nu_\epsilon(F_{v,b})$: feature shrinkage cannot lower Eq. (9). A zero block contributes loss one, and norms below ϵ are outside exact scale invariance. Clean anchoring and teacher-content retention provide the scene-dependent supervision complementary to this directional constraint.

B.1.4 NEA Gate and Stable-Subspace Projector

The lightweight gate G is a token-wise MLP shared across all patch tokens:

1. Linear(D_s, 96) + GELU
2. Linear(96, 1) + Sigmoid

The gate has $96D_s + 193$ parameters (37,057 for ViT-S/14). NEA learns $\Phi \in \mathbb{R}^{D_t \times r}$, with default $r = D_t/2$, and orthonormalizes its columns by QR retraction after each update. Deployment computes $\Phi^\top u_i^p$ directly, using $D_t r$ parameters and $O(N D_t r)$ multiply–adds without forming Q_Φ or back-projecting to D_t dimensions.

B.2 Algorithms

Algorithm A.1 gives the complete NoRA training pipeline. Algorithm A.2 expands the NEA gate–projection operator used in Stage 2 and deployment.

C Training Objectives and Optimization

C.1 Stage 1: Teacher Distillation

Stage 1 follows the global/local crop structure of DeFM (Patel et al., 2026) and distills its frozen backbone features. The objective below uses the released teacher features without requiring pretrained DINO/iBOT classification heads. Two 224×224 global crops use area fractions in $[0.4, 1.0]$; eight 98×98 local crops use $[0.1, 0.4]$. Crop aspect ratios lie in $[0.75, 1.33]$. The student and teacher receive corresponding geometry for spatial matching.

C.1.1 Global-Crop Distillation

For $G = 2$ clean global crops, the student global feature from one crop is matched to the frozen teacher feature from the other. Crop indices supplement the clean-view subscript s:

$$\mathcal{L}_{\text{Global}} = \frac{1}{G(G-1)} \sum_{i=1}^{G} \sum_{j \neq i} \left(1 - \left\langle \overline{u_{s,i}^g}, \overline{z_{s,j}^g} \right\rangle\right). \tag{14}$$

C.1.2 Local-Crop Distillation

Eight local crops ($L = 8$) of size 98×98 are processed by the student and matched to the teacher global crops:

$$\mathcal{L}_{\text{Local}} = \frac{1}{LG} \sum_{\ell=1}^{L} \sum_{g=1}^{G} \left(1 - \left\langle \overline{u_{s,\ell}^{g,\text{local}}}, \overline{z_{s,g}^g} \right\rangle\right). \tag{15}$$

Algorithm A.1 NoRA Training and Deployment Pipeline

Require: Clean simulated depth $\mathcal{D}_{\text{sim}}$, paired alignment set $\mathcal{D}_{\text{align}}$, simulated rollouts $\mathcal{D}_{\pi}^{\text{sim}}$
Require: Frozen teacher T, student encoder S_θ, heads $P_{\text{global}}, P_{\text{patch}}$, optional G_ϕ, Φ
Ensure: Frozen feature operator F_Ω and task head or policy π_ω

Stage 1: Simulation-guided teacher distillation

for each minibatch $\mathbf{X}_s \sim \mathcal{D}_{\text{sim}}$ **do**
  $(z_s^g, \{z_{s,i}^p\}_{i=1}^N) \leftarrow T(\psi(\mathbf{X}_s))$
  $(h_s^g, \{h_{s,i}^p\}_{i=1}^N) \leftarrow S_\theta(\psi(\mathbf{X}_s))$
  $u_s^g \leftarrow P_{\text{global}}(h_s^g)$; $u_{s,i}^p \leftarrow P_{\text{patch}}(h_{s,i}^p)$
  Minimize $\mathcal{L}_{\text{Distill}} = \mathcal{L}_{\text{Global}} + \mathcal{L}_{\text{Local}} + \mathcal{L}_{\text{Patch}} + \lambda_k \mathcal{L}_{\text{KoLeo}}$
  Update student parameters and EMA copy
end for
Initialize Stage 2 from the EMA student

Stage 2: Noise-resilient student alignment

for each batch $(\mathbf{X}_s, \mathbf{X}_r) \sim \mathcal{D}_{\text{align}}$ **do**
  $(z_s^g, \{z_{s,i}^p\}), (z_r^g, \{z_{r,i}^p\}) \leftarrow T(\psi(\mathbf{X}_s)), T(\psi(\mathbf{X}_r))$
  $(h_s^g, \{h_{s,i}^p\}), (h_r^g, \{h_{r,i}^p\}) \leftarrow S_\theta(\psi(\mathbf{X}_s)), S_\theta(\psi(\mathbf{X}_r))$
  Project h_s, h_r to u_s, u_r; compute s_i and y_i^ρ from Eq. (2)
  if using NIA **then**
    Set $w_i = 1$, $Q_\Phi = I_{D_t}$, and omit $\mathcal{R}_{\text{gate}}, \mathcal{R}_{\text{proj}}$
  else
    Obtain w, Q_Φ, paired downstream features, $\mathcal{R}_{\text{gate}}$, and $\mathcal{R}_{\text{proj}}$ from Algorithm A.2
  end if
  Minimize the selected NIA or NEA objective in Eq. (11)
  Update student, heads, and optional gate/basis; retract Φ; EMA-average student, heads, and gate
end for
$F_\Omega \leftarrow$ EMA student/heads/gate and final retracted basis; freeze F_Ω

Stage 3: frozen-encoder policy learning

for each rollout batch $\tau \sim \mathcal{D}_{\pi}^{\text{sim}}$ **do**
  Encode depth with F_Ω and train only π_ω using $\mathcal{L}_{\text{Policy}}$
end for
return F_Ω, π_ω

Algorithm A.2 Noise-Exclusive Alignment (NEA) Gate–Projection Operator

Require: Paired teacher tokens $\{z_{s,i}^p, z_{r,i}^p\}_{i=1}^N$, aligned student features $(u_v^g, \{u_{v,i}^p\})$, raw tokens h_s, h_r, gate G_ϕ, basis Φ

$Q_\Phi \leftarrow \Phi\Phi^\top$
for each patch token $i = 1, \ldots, N$ **do**
  $s_i \leftarrow \exp(-\|\bar{z}_{s,i}^p - \bar{z}_{r,i}^p\|_2/\tau_s)$; derive shared target y_i^ρ
  $m_{s,i} \leftarrow G_\phi(h_{s,i}^p)$; $m_{r,i} \leftarrow G_\phi(h_{r,i}^p)$; $w_i \leftarrow (m_{s,i} + m_{r,i})/2$
  $a_{v,i} \leftarrow \Phi^\top u_{v,i}^p$ for $v \in \{s, r\}$
end for
$a_v^g \leftarrow \Phi^\top u_v^g$ for $v \in \{s, r\}$
$c(a_v) \leftarrow N^{-1} \sum_i a_{v,i}$; $\tilde{a}_{v,i} \leftarrow m_{v,i} a_{v,i} + (1 - m_{v,i}) c(a_v)$
Compute $\mathcal{R}_{\text{proj}}$ from paired teacher tokens and s using Eq. (4)
Construct $F_{\theta,\phi,\Phi}(\mathbf{X}_s)$ and $F_{\theta,\phi,\Phi}(\mathbf{X}_r)$ from the context-filled low-rank coordinates
Compute blockwise normalized consistency from Eq. (9)
Compute $\mathcal{R}_{\text{gate}}$ from Eq. (3)
return $w, Q_\Phi, F_{\theta,\phi,\Phi}(\mathbf{X}_s), F_{\theta,\phi,\Phi}(\mathbf{X}_r), \mathcal{R}_{\text{gate}}, \mathcal{R}_{\text{proj}}$

C.1.3 PATCH-LEVEL DISTILLATION

Patch distillation uses unmasked global crops. Student patch tokens are projected to the teacher dimension and matched to the corresponding teacher patch tokens:

$$\mathcal{L}_{\text{Patch}} = \frac{1}{GN} \sum_{g=1}^{G} \sum_{i=1}^{N} \text{SmoothL1}\left(\overline{u^p_{s,g,i}}, \overline{z^p_{s,g,i}}\right). \tag{16}$$

C.1.4 KOLEO REGULARIZER

The KoLeo regularizer (Sablayrolles et al., 2019) encourages diversity across the B sample descriptors in a minibatch. Here $h^g_{s,b}$ is the raw student global descriptor for sample b (with a small numerical floor inside the logarithm):

$$\mathcal{L}_{\text{KoLeo}} = -\frac{1}{B} \sum_{b=1}^{B} \log \min_{b' \neq b} \|h^g_{s,b} - h^g_{s,b'}\|_2 \tag{17}$$

The total loss is Eq. (1), with $\lambda_k = 0.1$. Its EMA student initializes Stage 2 and defines Pure-Sim.

C.2 STAGE 2: NOISE-RESILIENT ALIGNMENT

C.2.1 SPATIAL GATE OBJECTIVE

The target y^ρ_i selects the top $\lfloor \rho N \rfloor$ teacher-stability scores. Since $\exp(-d_i/\tau_s)$ is monotone in d_i, τ_s changes the continuous projection weights but not the top-k ranking. Each single-view gate learns the shared paired target during training.

In Eq. (3), $\text{BCE}(m, y) = -y \log m - (1 - y) \log(1 - m)$ and $H_{\text{Bern}}(w) = -w \log w - (1 - w) \log(1 - w)$; probabilities are clipped to $[10^{-6}, 1 - 10^{-6}]$. Teacher features and stability targets are detached; the gate receives its supervised and alignment gradients as detailed in App. C.2.3.

C.2.2 STABLE SUBSPACE AND DEPLOYMENT OPERATOR

The two terms in Eq. (4) suppress paired teacher displacement and retain stable teacher content, respectively. QR retraction fixes basis scale and rank; the learned basis is shared across inputs and fixed at deployment.

Context filling in Eq. (5) interpolates each projected token with the all-token mean. It preserves the spatial grid but does not certify that mean as reliable. Downstream tasks use the r-dimensional coordinates and keep their aggregation $\mathcal{A}$ fixed across alignment, readout fitting, and deployment.

C.2.3 FEATURE MATCHING AND CONFIGURATION

The feature-matching loss combines angular agreement and a coordinate-wise discrepancy between normalized features:

$$\ell_{\text{match}}(a, b) = \lambda_{\cos}(1 - \langle \bar{a}, \bar{b} \rangle) + \lambda_{\text{l1}} \, \text{SmoothL1}(\bar{a}, \bar{b}). \tag{18}$$

Matching uses $\hat{u} = Q_\Phi u$ and $\hat{z} = Q_\Phi z$, with frozen teacher features. Gradients reach both student branches, the gate through token weights and exported outputs, and Q_Φ on both matching sides. The token loss divides by total spatial weight plus ϵ.

SmoothL1 is averaged over feature coordinates and all objectives over the minibatch. Direction weights α and $1 - \alpha$ sum to one. Deployment consistency averages uniformly over normalized blocks. The matching coefficients are:

$$\lambda_{\cos} = 0.9, \qquad \lambda_{\text{l1}} = 0.1. \tag{19}$$

The unprojected clean anchor supervises u_s before gating or projection, while cross-view matching supplies clean-teacher targets to u_r. Both views share the backbone and alignment heads. We use $\alpha = 2/3$ and $\lambda_{\text{scene}} = \lambda_{\text{token}} = 2$, so noisy-to-clean matching has twice the reverse-direction weight. The one-way control changes only α (App. E.3).

C.2.4 FULL-SUPPORT AND SELECTIVE OBJECTIVES

NIA sets $m_{v,i} = w_i = 1$ and $\Phi = Q_\Phi = I_{D_t}$, omitting context mixing and both operator regularizers. NEA learns the gates and rank-r basis and includes $\mathcal{R}_{\text{gate}}$ and $\mathcal{R}_{\text{proj}}$. Both use Eq. (10), the same Stage-1 initialization, and an unprojected clean anchor.

C.2.5 NIA AND NEA HYPERPARAMETERS

NIA and NEA share the alignment coefficients below; the right-hand settings apply only to NEA. Their common training schedule is specified in App. C.3.

Table A.2: Alignment and operator hyperparameters.

Shared parameter	Value	NEA parameter	Value
α	$2/3$	λ_{gate}	1.0
λ_{scene}	2.0	λ_{proj}	1.0
λ_{token}	2.0	λ_{keep}	1.0
λ_{sem}	1.0	λ_{gc}	0.1
λ_{dep}	1.0	λ_b	0.1
Cosine/SmoothL1 weights	0.9 / 0.1	λ_e	0.05
Effective batch	64 pairs	τ_s	0.5
Learning rate	10^{-4}	ρ	0.8
Optimizer updates	30,000	r	$D_t/2$
Warmup updates	5,000	Basis learning rate	5×10^{-5}
EMA decay	0.9998	Basis constraint	QR retraction
Weight decay	0.05	Gate target	Top-k

C.3 TRAINING THE NORA FEATURE OPERATOR

AdamW uses $(\beta_1, \beta_2) = (0.9, 0.999)$, weight decay 0.05 on encoder weights, no decay on biases, normalization parameters or Φ, and gradient clipping at norm 1. Stage 1 runs for 100 epochs with batch 64, learning rate 2×10^{-4} and five warmup epochs. Stage 2 uses 30,000 updates, batch 64 pairs, learning rate 10^{-4} for the encoder/heads/gate and 5×10^{-5} for the basis, 5,000 warmup updates and cosine decay to 10^{-6}. EMA decay is 0.9998 for the student, heads and gate; the basis uses its final QR-retracted state.

Stage 2 starts from the Stage-1 EMA and uses the terminal scheduled checkpoint. Downstream readouts are selected only on their development splits; robot outcomes do not select perception checkpoints.

D EXPERIMENTAL SETTINGS AND EVALUATION PROTOCOLS

Numerical Scope. The numerical tables are conditional estimates for raw-output MSE consistency; they do not evaluate the normalized objective in Eq. (9).

Training Roles and Data Provenance. Stage 1 distills rendered depth; Stage 2 aligns LingBot-Depth/ByteCameraDepth reference–sensor pairs. The dataset inset in Fig. 2 supplies Stage-2 observations. Isaac Sim provides clean-depth policy demonstrations, while PRISM-corrupted ManiSkill3 sequences are used only for the temporal probe.

D.1 DATA, BASELINES, AND COMPARISON SCOPE

D.1.1 ALIGNMENT DATA AND CORRESPONDENCE

Stage 2 draws on the public LingBot-Depth (Tan et al., 2026) and ByteCameraDepth (Liu et al., 2026) RGB-D datasets, using registered reference-depth/raw-depth observations for alignment. The training-sensor pool covers nine camera models: Orbbec Gemini 335/335L, RealSense

D405/D415/D435/D455/L515, Microsoft Azure Kinect DK, and Stereolabs ZED2i. LingBot-Depth provides Gemini 335/335L and RealSense D405/D415/D435/D455 data; ByteCameraDepth additionally covers L515, Azure Kinect, and ZED2i. Zed2i-N denotes the neural depth mode of ZED2i rather than a separate camera model. We split source scenes into 70% training, 15% validation, and 15% evaluation, keeping an entire capture sequence and its reconstructed scene in one split. Near-duplicate views and object instances from a sequence do not cross these boundaries. All methods receive the same source scene split, while their use of paired indices and RGB is stated in App. D.1.2.

LingBot-Depth supplies the paired reference/raw-depth observations used for alignment. For ByteCameraDepth, we use Depth Anything 3 (Lin et al., 2025a) for offline alignment and denoising to construct a clean-reference counterpart to each real noisy depth observation. The resulting references are aligned to the observed depth view before paired feature supervision. Here *clean-reference* denotes the denoised training target, including model-generated pseudo-reference depth; it does not imply independently measured ground truth or simulator-rendered depth. Teacher residuals can therefore reflect sensor perturbations, reference-estimation error, and registration error together; the gate learns the resulting paired stability cue. Depth Anything 3 is used only to prepare alignment data, not during policy training or deployment. Stage 1 and simulation-only policy learning instead use rendered simulated depth. Thus the no-external-restoration claim concerns the deployed feature-to-action pipeline.

The validity support is derived from reference geometry and registration, not from the NoRA gate. A token is eligible for paired token losses when at least half its pixels have valid correspondences. For partially registered pairs, the token sums and N-based averages in the stability, gate, projection, and clean patch-anchor losses are restricted to this eligible set, and top-ρ uses its size. This is the masked implementation of the fully corresponding setting in Sec. 3.3. Pairs with no eligible tokens are excluded from alignment. The exported grid, all-token context mean, and global features still use the complete image; eligibility never removes tokens from the deployed operator. Deployment consistency uses the complete exported outputs, as in Eq. (9), rather than a model-selected region. Missing raw returns remain part of the observed input, including when the corresponding reference pixel is valid. Random resizing, cropping, and horizontal flips are shared between paired views. The depth scale is preserved.

D.1.2 Baseline Configurations

Tab. A.3 specifies model architectures and inputs. CNN classification uses ResNet-18/34/50 and EfficientNet-B2/B4/B6; each compares Pure-Sim, adapted DeFM, Theia-CNN, NIA and NEA. Theia-CNN is our CNN-student adaptation of Theia. ViT comparisons retain native patch sizes; DINOv3 uses LVD-1689M checkpoints, V-JEPA 2.1 uses its distilled L/16 model, and DeFM-L/14 supplies the frozen teacher for every NoRA backbone.

Predictive comparisons use ViT-L models; Theia and PriorDA lack the corresponding released student or conditional-refinement checkpoint. Robot comparisons retain Theia-B/16 and PriorDA-B/14 as exceptions. All model names denote mixed-data-adapted configurations; Pure-Sim is the initialization control.

Mixed-Data Adaptation for All Baselines. All eight baseline families and NoRA are adapted independently on identical ByteCameraDepth/LingBot-Depth scene splits, registered pairs, sample order and augmentation draws. This includes complete robot restoration systems and predictive feature operators. Sampling and optimization are specified in App. D.2.

Depth Input Mapping. DeFM and NoRA use the official metric-aware three-channel transform, including its released mid-range normalization convention. Invalid depth remains zero; unit conversion precedes normalization. RGB-pretrained models receive per-image min–max scaled valid depth repeated across three channels and their checkpoint's normalization, following DeFM's RGB-baseline adaptation. Physical depth is capped at 100m before mapping for all models; inferred inverse depth uses the conversion in App. D.2.1. Restoration-pretrained predictive operators instead retain the method-specific mappings in App. D.3.1. Masked prediction computes any content-dependent input statistics from visible pixels only. Thus the full-image target cannot leak through a min/max value. For per-image min–max mapping, a constant valid depth map has zero scaled values before checkpoint normalization; DeFM/NoRA retain its metric value in their fixed-scale channels. Invalid

Table A.3: Baseline Scope. C: CNN classification; S: ViT segmentation; P: predictive probes; R: robot transfer. Model sizes retain documented architectures; † denotes our CNN implementation of Theia, not a released public CNN checkpoint.

Family	Configuration	Track	Input to encoder/system
DeFM	Six CNNs; S/14; L/14	C/S/P/R	Metric depth
Theia	CNN† (C); S/16 (S); B/16 (R), cdiv	C†/S/R	Replicated depth
DINOv3	S/16 (S); L/16 (S/P/R)	S/P/R	Replicated depth
V-JEPA 2.1	L/16	S/P/R	Replicated depth
CDM	L/14 depth (P); RGB-D (R)	P/R	Depth (P); RGB-D (R)
PromptDA	L/14 + prompted decoder	P/R	Depth + constant RGB (P); RGB-D (R)
PriorDA	B/14 coarse + conditional	R	Native RGB-D
LingBot-MDM	L/14 depth (P); RGB-D (R)	P/R	Depth (P); RGB-D (R)

pixels are masked to zero after preprocessing, including entirely invalid inputs. Such inputs remain in the evaluation if labels are valid.

Theia exports its cdiv student features; teacher translators are excluded at deployment. V-JEPA 2.1 uses its image tokenizer for single frames. Native backbone widths and patch sizes are preserved during adaptation.

D.1.3 Restoration-Based Robot Pipelines

Robot evaluation uses mixed-data-adapted CDM, PromptDA, PriorDA, and LingBot-MDM (Liu et al., 2026; Lin et al., 2025b; Wang et al., 2025; Tan et al., 2026) to restore observed RGB-D into depth, followed by a separate DeFM-L/14 encoder and DP3. Each restoration network retains its required fusion, depth-output heads, and postprocessing during adaptation. All four cascades use the same separately mixed-data-adapted DeFM-L/14 feature checkpoint, with no additional cascade-specific encoder-update budget. After adaptation, the complete restoration-to-feature pipeline is frozen and identical during simulation policy fitting and real inference. Each method has independently fitted policy weights under the common architecture and optimization budget.

CDM uses two ViT-L branches; PromptDA and LingBot-MDM use ViT-L; PriorDA uses v1.1 ViT-B coarse and conditional models. Their native encoders, fusion/prompt modules and depth heads are adapted with registered reference-depth targets for 30,000 updates. The terminal restoration checkpoint is fixed across cameras. Predictive feature-only configurations are specified separately in App. D.3.1.

D.2 Frozen Visual Transfer

Shared Mixed-Data Adaptation before Freezing. Each adaptation update samples 32 pairs per source, uniformly over scenes and then frames with replacement, giving global batch 64 and 1.92 million pair presentations. Methods share sample order, correspondence masks and geometric transforms. RGB, noisy depth and reference depth undergo the same crop before method-specific resizing. Visual/predictive operators use 224×224 inputs; complete restoration systems retain native resolutions (App. D.4.3). Optimization follows App. C.3, with gradient accumulation as needed for the shared effective batch.

Representation baselines and the depth-only restoration feature operators use the full-support form of Eq. (10): $m_i = 1$, $\Phi = I$, and the NIA coefficients in App. C.2.5. Each such baseline retains its own backbone and uses linear–LayerNorm global and patch alignment heads to connect to the same frozen, original DeFM-L/14 teacher. No learned gate or subspace selection is added to these baselines. NoRA uses its NEA objective on exactly the same pairs. Complete RGB-D restoration systems follow the restoration objectives in App. D.1.3 under the same mixed-data schedule. This distinguishes matched adaptation data from method-specific supervision. The original teacher remains separate from the adapted DeFM baseline and cascade feature checkpoint. For different patch sizes, baseline tokens are interpolated onto the teacher grid only for alignment supervision; downstream segmentation retains native token grids.

Terminal EMA backbones, heads, restoration modules and normalization statistics are frozen; NEA also freezes its gate and basis. Full-support outputs have 1024 channels and NEA has 512. Pure-Sim omits adaptation, while other internal controls retain their specified interventions.

Only adaptation updates the feature operators. Classification, segmentation and predictive heads are fitted separately after freezing; predictive heads are not reused by robot policies. Evaluation scenes are excluded from adaptation and predictor fitting. Adaptation uses no real-robot actions or task-success labels.

Shared Readout Settings. Frozen linear classification, kNN and tokenwise segmentation follow DeFM (Patel et al., 2026). Linear probes use AdamW for 50 epochs, five warmup epochs, cosine decay and seeds $\{0, 1, 2\}$. Development splits select learning rate from $\{10^{-4}, 3 \times 10^{-4}, 10^{-3}\}$ and weight decay from $\{10^{-4}, 10^{-2}\}$. Features are normalized and encoder inference is deterministic; samples and augmentations are shared across methods.

D.2.1 Image Classification on CNNs

Following DeFM (Patel et al., 2026), frozen Depth-Anything-V2-L (Yang et al., 2024) converts ImageNet RGB to relative inverse depth. DeFM/NoRA use its inverse-depth-to-pseudo-metric transform; other encoders use the mapping in App. D.1.2. RGB and class labels do not enter depth conversion or encoder evaluation. A seeded, class-stratified 90/10 split of ImageNet training images selects probe settings; readouts are then refitted on the full training set and tested on the official 50,000-image validation set.

Additional CNN Comparisons. Theia-CNN uses ResNet or EfficientNet students with Theia's cdiv teachers (CLIP, DINOv2 and supervised ViT) and feature-matching objective (Shang et al., 2025). Spatial features are resampled for teacher-specific translators. This distillation precedes the shared depth adaptation; only the student and depth-alignment heads are exported. These are custom CNN adaptations, not released Theia CNN checkpoints.

NoRA-NIA shares the CNN architecture, Stage-1 checkpoint and adaptation schedule with NEA, but uses full spatial support and the identity basis. Newly fitted linear/kNN heads compare their frozen representations in Tab. A.5.

Linear and kNN Readouts. The input canvas is 224×224. Each classifier is a single linear layer on the frozen global descriptor, including alignment heads and NoRA's stable-subspace projection. Cross-entropy trains only the classifier, with batch 64 and equal class weights under the common schedule above. Training uses random resized crops with scale 0.8–1.0 and horizontal flips; evaluation uses deterministic resizing and a center crop. kNN uses normalized features, cosine similarity, $k = 20$, and temperature 0.07, with training features only in the retrieval bank. We report linear Top-1 and kNN Top-1/Top-5 accuracy separately for all six CNN architectures.

D.2.2 Semantic Segmentation on ViTs

Following DeFM (Patel et al., 2026), the ViT benchmark spans indoor ScanNet (Dai et al., 2017) and SUN-RGBD (Song et al., 2015), outdoor OFFSED (Neigel et al., 2021), and tabletop GraspNet-1B (Fang et al., 2020). Tab. A.4 records the acquisition hardware. Structure Sensor, Kinect v1, and Xtion use structured light; Kinect v2 and Azure Kinect use time of flight; OFFSED uses ZED stereo depth. SUN-RGBD also contains RealSense captures. For GraspNet we select the Azure Kinect stream to follow DeFM's segmentation setting, rather than pooling its Kinect and RealSense streams. These benchmarks retain native noisy depth and missing returns; no restoration or reference-depth replacement is applied to the input.

Each dataset uses its own class vocabulary and linear head. Official scene-disjoint evaluation partitions are preserved where available, with 10% of training scenes reserved for model selection. When no official scene-disjoint split is supplied, a seeded 70/15/15 scene or sequence partition is fixed before feature extraction. Adjacent frames stay together.

All encoders use the same dataset-specific label maps and depth/label coordinates. Labeled pixels remain eligible when input depth is invalid; only unlabeled ground truth is ignored.

Dense Readout and Optimization. A single tokenwise linear classifier maps frozen features to the dataset's semantic labels. All encoders use a 224×224 canvas and retain their native patch grids;

Table A.4: Visual-Transfer Datasets and Acquisition Hardware. Class counts follow DeFM and the stated benchmark label spaces.

Dataset	Sensor / depth source	Classes	Domain
ImageNet-1K-Depth	Depth Anything V2 (inferred)	1000	Objects
ScanNet	Structure Sensor	40	Indoor
SUN-RGBD	Kinect v1/v2, RealSense, Xtion	37	Indoor
OFFSED	ZED stereo camera	5	Outdoor
GraspNet-1B	Azure Kinect	40	Tabletop

NoRA also retains its frozen gate and context operator. Bilinear interpolation restores logits to the original label resolution, without an RGB guide, learned multi-scale decoder, or encoder update. Training uses batch 32, pixelwise cross-entropy, equal class weights, and the common schedule above. Depth and labels share aspect-preserving resizing, padding, and horizontal flips; padding is removed before scoring.

We report mIoU over non-ignored classes; Avg. is the equal-weight mean across four datasets. Scoring support is independent of the predicted gate.

D.2.3 Additional Visual Transfer Results

Tab. A.5 gives CNN classification results. The segmentation examples below share scenes and label colors across methods.

Table A.5: **Frozen-Backbone Depth Classification on CNNs.** ImageNet-1K-Depth *accuracy* (%, ↑). *Linear* denotes *Top-1*; *kNN-1/5* denote *Top-1/5*. † denotes CNN distillation based on the official Theia code.

ResNet					EfficientNet				
Backbone	Method	Linear	kNN-1	kNN-5	Backbone	Method	Linear	kNN-1	kNN-5
ResNet-18	Pure-Sim	47.8	42.9	66.8	EfficientNet-B2	Pure-Sim	47.6	44.7	68.6
ResNet-18	Theia†	45.9	39.8	63.7	EfficientNet-B2	Theia†	44.9	41.5	65.9
ResNet-18	DeFM	50.7	45.8	69.8	EfficientNet-B2	DeFM	50.5	47.8	71.7
ResNet-18	NoRA-NIA	50.4	45.4	69.5	EfficientNet-B2	NoRA-NIA	**50.6**	47.6	71.6
ResNet-18	**NoRA**	**50.9**	**46.6**	**70.4**	EfficientNet-B2	**NoRA**	50.3	**48.2**	**72.0**
ResNet-34	Pure-Sim	51.2	45.2	69.5	EfficientNet-B4	Pure-Sim	51.5	47.1	71.4
ResNet-34	Theia†	49.6	43.0	67.2	EfficientNet-B4	Theia†	50.2	45.6	69.8
ResNet-34	DeFM	54.2	48.0	72.5	EfficientNet-B4	DeFM	54.6	50.1	74.5
ResNet-34	NoRA-NIA	54.3	48.3	72.6	EfficientNet-B4	NoRA-NIA	54.9	50.4	74.8
ResNet-34	**NoRA**	**55.1**	**49.3**	**73.5**	EfficientNet-B4	**NoRA**	**55.4**	**51.2**	**75.4**
ResNet-50	Pure-Sim	58.3	50.1	74.6	EfficientNet-B6	Pure-Sim	55.8	49.8	74.4
ResNet-50	Theia†	56.8	48.4	72.9	EfficientNet-B6	Theia†	54.6	48.1	72.8
ResNet-50	DeFM	61.1	53.2	77.3	EfficientNet-B6	DeFM	58.9	52.9	77.4
ResNet-50	NoRA-NIA	61.3	53.6	77.6	EfficientNet-B6	NoRA-NIA	59.1	53.3	77.6
ResNet-50	**NoRA**	**62.2**	**54.8**	**78.3**	EfficientNet-B6	**NoRA**	**59.6**	**54.2**	**78.2**

Figures A.1–A.4 provide additional qualitative segmentation comparisons on ScanNet, SUN-RGBD, OFFSED, and GraspNet-1B, respectively.

D.3 Predictive Representation Protocols

D.3.1 Masked Spatial Feature Completion

Transferred Encoders and Common Predictor. All feature operators in Tab. 2 are frozen after the shared adaptation. The V-JEPA 2.1-style predictor (Mur-Labadia et al., 2026) has 24 blocks, width 384, 12 attention heads and MLP ratio four. Its input width is 1024; NoRA's 512-dimensional coordinates are lifted with frozen Φ. Predictor/adapter initialization, budgets and target dimensions are shared.

All inputs use a 224×224 image canvas. After masked-input encoding, native depth token grids are deterministically mapped to a common 16×16 grid; masks are defined in image coordinates before encoding, including for patch-16 models. A shared frozen V-JEPA 2.1-L image encoder supplies the complete RGB context for spatial completion. Modality and position embeddings join its native RGB grid with visible depth tokens.

Restoration-Pretrained Feature Interfaces. CDM (Liu et al., 2026) exports final-layer ViT-L/14 depth tokens before cross-modal fusion and decoding. Its depth branch and alignment heads receive shared adaptation. Valid metric depth is converted to inverse depth and repeated across three channels; invalid pixels are zero. The RGB branch and depth decoder are omitted for prediction, while robot evaluation uses the complete RGB-D system.

LingBot-MDM (Tan et al., 2026) uses LingBot-Depth-v0.5 ViT-L/14 tokens before output projections and decoding. RGB is fixed to the checkpoint image mean during adaptation and prediction; depth retains its metric/log mapping. Internal stochastic masking is disabled, leaving only the external spatial mask. This defines the depth-only predictive configuration.

PromptDA (Lin et al., 2025b) exports the final 256-channel depth-prompted DPT refinement map before output convolutions. RGB is fixed to the checkpoint image mean. The map is resampled and projected to 1024 dimensions; the backbone, prompt-fusion path and projection receive the shared full-support adaptation and are then frozen.

For all three restoration feature operators, the backbone, depth-conditioning modules, and alignment heads are adapted on the same paired training scenes and frozen before predictor fitting. Spatial completion masks depth before any feature operator sees it, then supplies full RGB through the same independent context branch used for all methods. Temporal forecasting processes each observed depth frame independently, without RGB, future depth, or restoration of a future frame. No restoration feature operator obtains additional temporal observations.

Data, Masking, and Information Access. The spatial task uses synchronized, registered RGB and real noisy depth from LingBot-Depth (Tan et al., 2026). Its training/validation/test scenes follow the adaptation partition in App. **D.1.1**; ByteCameraDepth does not supply spatial test images. Artificial masks are sampled uniformly without replacement on the depth patch grid, hiding exactly 64, 128, or 192 of its 256 patches. Training samples the three ratios with equal probability. Testing uses five fixed masks per image and ratio, with identical image/mask IDs for all models. These artificial masks are separate from naturally missing sensor returns.

Masked pixels are set to the invalid-depth value before normalization, encoding, or NoRA's context filling. Any content-dependent normalization statistics, including PromptDA prompt-depth scaling, use visible valid depth only. The same statistics transform the visible prompt, while artificial and sensor-invalid locations retain their invalid mask. The predictor receives depth tokens only at visible positions, full registered RGB tokens, and positional queries for masked depth locations. Neither a full-depth global descriptor nor full-depth features enter its context. Since complete RGB is deliberately available, this experiment measures cross-modal feature completion, not depth-only reconstruction. The reported comparison holds RGB context fixed across methods; it does not isolate the incremental contribution of depth relative to RGB alone.

Common Targets and Scoring. Let T denote the original, frozen DeFM-L/14 reference encoder, distinct from the adapted DeFM baseline. The target at patch i is $z_i = \mathrm{sg}(\mathrm{norm}(T(D)_i))$, computed from the complete noisy depth frame. All models predict this same 1024-dimensional target. Thus the cosine-similarity endpoint compares predictions in a common representation, rather than incomparable native feature spaces. Full noisy depth is not a clean geometric reference: the spatial endpoint measures completion, and does not by itself establish denoising accuracy. Because this reference is a DeFM model and NoRA's teacher, the endpoint is interpreted specifically as recovery of that representation; independent visual-transfer and robot endpoints establish broader utility.

For masked set M and unit-normalized prediction $\bar{z}_i$, training minimizes

$$\mathcal{L}_{\mathrm{mask}} = \frac{1}{|M|d} \sum_{i \in M} \|\bar{z}_i - z_i\|_1, \qquad d = 1024. \tag{20}$$

We also supervise visible depth positions with the same mean normalized L1 loss, weighted by 0.7 for spatial training. This uniform context term is an explicit simplification of V-JEPA 2.1's

distance-weighted dense supervision. Main-table scoring uses only artificially masked positions whose original depth has at least 50% valid returns; the eligibility mask is fixed independently of the model and gate. Masks still cover all patch positions, including naturally missing patches. Cosine similarity is averaged over eligible tokens, then masks and frames, and finally scenes. The reported endpoint is shared-target cosine similarity. It measures access to the reference representation under the stated conditioning, rather than physical depth accuracy.

D.3.2 Multi-Step Temporal Feature Prediction

ManiSkill3 Trajectories and PRISM Noise Augmentation. ManiSkill3 (Tao et al., 2025) on SAPIEN (Xiang et al., 2020) supplies trajectories; articulated tasks use PartNet-Mobility (SAPIEN Team, 2026). A fixed external camera renders ideal depth for PushCube-v1, PickCube-v1, StackCube-v1, PegInsertionSide-v1 and OpenCabinetDrawer-v1. Entire trajectories, object instances and scene seeds are assigned to 70/15/15 training/validation/test groups within each task before window sampling.

A fixed PRISM (Liang et al., 2026) checkpoint augments simulated depth with realistic sensor noise. Its required rendering inputs are retained in the generator, while only the resulting depth enters the temporal model. Sensor configuration is fixed within an episode, and the checkpoint, noise settings, and random seeds are identical across encoder comparisons. Both the noisy sequence and its original ideal depth are cached. Framewise noise synthesis does not enforce temporal consistency; the evaluation therefore concerns the generated noisy sequences rather than a calibrated model of temporal sensor noise. PRISM is not optimized on validation/test trajectories.

Observation Window and Prediction Horizons. Eight noisy depth frames $D^n_{t-7:t}$ sampled at 10 Hz predict horizons $h \in \{1, 8, 16, 24, 32\}$, or 0.1/0.8/1.6/2.4/3.2 seconds. Windows with resets, dropped frames or episode termination are excluded; frames are not duplicated. RGB, proprioception and future actions are not predictor inputs.

Historical frames are encoded independently by the frozen depth operator. The predictor retains the spatial architecture above, with temporal positions and future-time mask queries but no RGB branch. Frame-causal attention prevents history tokens from accessing future queries and orders the query blocks by horizon. All five target maps are predicted directly from the available history and learned queries; no ground-truth future feature is fed back. The reference target is $z_{t+h,i} = \mathrm{sg}(\mathrm{norm}(T(D^c_{t+h})_i))$ from ideal future depth, where D^c is saved by the simulator. This endpoint tests forecasting of underlying scene structure from noisy observations, rather than forecasting a random future sensor-noise realization.

The prediction loss averages normalized L1 equally over the five horizons and their valid target tokens. A context loss on historical tokens uses ideal historical targets with weight 0.5. These weights and the direct forecast schedule are shared across all encoders. Tab. 2 reports cosine similarity at every horizon; Tab. A.7 breaks down the same metric by task. Metrics average over windows within episodes and then equally over episodes and tasks. All predictions use observed history only. Whole-frame feature similarity combines static and moving content; this endpoint alone does not isolate motion forecasting or the value of multiple historical frames.

Predictor Optimization and Reporting. Spatial and temporal predictors are trained separately for 30,000 AdamW updates, global batch 64, learning rate 10^{-4}, weight decay 0.05, 3,000 warmup updates, cosine decay to 10^{-6}, and gradient clipping at norm 1. All backbones, normalization statistics, NoRA operators, RGB encoders, and reference targets remain frozen. Seeds $\{0, 1, 2\}$ share training samples, masks, initialization, and PRISM realizations across methods. The final-update predictor is evaluated without test-based checkpoint selection. The prediction tables report cosine-similarity point estimates; differences are descriptive and are not presented as statistical significance tests. PCA visualizations use one basis fitted to reference training features and fixed across predictions and targets. These heads measure predictive accessibility and are discarded before the independently trained robot-control evaluation.

D.3.3 Additional Predictive Results

The main table covers all eligible ViT-L encoders, while Fig. 4 focuses on DeFM-L and NoRA-L for a readable qualitative comparison. Tables A.6 and A.7 break down the reported cosine similarity

by scene category and task at every mask ratio and forecast horizon. These use the same adapted encoder checkpoints, predictor training protocol, and shared reference targets as the main table. Spatial completion retains full RGB context; temporal forecasting uses eight observed depth frames without future actions.

Table A.6: **Spatial Completion by Scene Category.** Shared-target *cosine similarity* (↑). The three scene categories receive equal weight in Tab. 2.

Encoder	Mask Ratio		
	0.25	0.50	0.75
Broad continuous surfaces			
PromptDA	0.963	0.950	0.932
CDM	0.972	0.960	0.944
LingBot-MDM	0.981	0.978	**0.974**
DINOv3	0.980	0.969	0.951
V-JEPA 2.1	0.977	0.967	0.952
DeFM	**0.986**	0.978	0.961
NoRA	0.984	**0.980**	0.968
Clutter and mutual occlusion			
PromptDA	0.918	0.894	0.859
CDM	0.938	0.916	0.884
LingBot-MDM	0.955	0.948	**0.943**
DINOv3	0.949	0.927	0.895
V-JEPA 2.1	0.945	0.928	0.899
DeFM	**0.962**	0.945	0.919
NoRA	0.960	**0.952**	0.934
Thin structures and depth discontinuities			
PromptDA	0.889	0.858	0.818
CDM	0.911	0.884	0.845
LingBot-MDM	0.938	**0.929**	**0.913**
DINOv3	0.924	0.901	0.861
V-JEPA 2.1	0.921	0.898	0.866
DeFM	**0.941**	0.919	0.883
NoRA	0.935	0.925	0.904

D.4 Robot Manipulation Protocols

D.4.1 Simulation Training and Real Transfer

Simulator and Demonstrations. Isaac Sim (NVIDIA, 2026) provides shared Game, Canteen and Kitchen demonstrations using the RealMan RM75-6F arm, Pika gripper and calibrated workspace geometry. Entire trajectories follow an 80/10/10 split, with shared rendering and initial-state randomization. Policies train on clean depth without real-action supervision. DeFM's published manipulation protocol instead augments depth with speckle, Gaussian, dropout and stick noise (Patel et al., 2026); Ex.3 and the PRISM temporal probe do not test policy-training augmentation.

Cartesian Control and Proprioception. The action is a 10-dimensional absolute target: TCP translation in the robot base frame (3), a continuous two-column rotation representation (6) (Zhou et al., 2019), and normalized gripper opening (1). The rotation is projected to $SO(3)$ before command execution; the calibrated opening range maps the gripper scalar to physical jaw separation. This Cartesian action space avoids forcing policies to choose among redundant RM75 joint configurations. Proprioception contains measured joint positions (7), TCP translation/rotation (9), and gripper opening (1), totaling 17 dimensions; the common visual-plus-proprioceptive observation therefore has 273 dimensions per frame. Force/torque signals are not policy inputs in this comparison.

Table A.7: **Temporal Prediction by Task.** Shared-target *cosine similarity* (↑). The five tasks receive equal weight in Tab. 2.

Encoder	Future Steps				
	1	8	16	24	32
PushCube-v1					
PromptDA	0.935	0.849	0.744	0.659	0.615
CDM	0.949	0.874	0.778	0.694	0.651
LingBot-MDM	0.972	0.927	0.884	0.856	**0.838**
DINOv3	0.958	0.892	0.794	0.716	0.665
V-JEPA 2.1	0.954	0.889	0.802	0.731	0.679
DeFM	0.965	0.917	0.863	0.828	0.802
NoRA	**0.976**	**0.931**	**0.889**	**0.859**	0.834
PickCube-v1					
PromptDA	0.918	0.798	0.678	0.579	0.529
CDM	0.936	0.826	0.711	0.613	0.551
LingBot-MDM	0.965	0.897	0.839	0.798	0.764
DINOv3	0.946	0.843	0.719	0.627	0.579
V-JEPA 2.1	0.945	0.847	0.727	0.638	0.581
DeFM	0.955	0.876	0.807	0.758	0.724
NoRA	**0.969**	**0.908**	**0.853**	**0.807**	**0.769**
StackCube-v1					
PromptDA	0.903	0.764	0.633	0.528	0.469
CDM	0.923	0.793	0.664	0.557	0.491
LingBot-MDM	0.958	0.867	0.812	0.763	**0.728**
DINOv3	0.937	0.809	0.678	0.572	0.513
V-JEPA 2.1	0.938	0.817	0.693	0.587	0.524
DeFM	0.947	0.844	0.767	0.718	0.681
NoRA	**0.963**	**0.884**	**0.819**	**0.768**	0.712
PegInsertionSide-v1					
PromptDA	0.898	0.823	0.713	0.639	0.604
CDM	0.924	0.857	0.752	0.677	0.638
LingBot-MDM	**0.965**	0.921	**0.882**	**0.851**	**0.829**
DINOv3	0.934	0.876	0.779	0.699	0.649
V-JEPA 2.1	0.931	0.871	0.776	0.707	0.656
DeFM	0.948	0.901	0.852	0.821	0.793
NoRA	0.961	**0.922**	0.881	0.847	0.826
OpenCabinetDrawer-v1					
PromptDA	0.908	0.778	0.654	0.563	0.508
CDM	0.931	0.812	0.694	0.591	0.533
LingBot-MDM	0.963	**0.879**	**0.815**	**0.781**	**0.759**
DINOv3	0.941	0.832	0.703	0.618	0.567
V-JEPA 2.1	0.939	0.836	0.718	0.626	0.572
DeFM	0.951	0.861	0.794	0.746	0.706
NoRA	**0.966**	0.868	0.805	0.779	0.751

Predicted waypoints are spaced at 10 Hz. A separate 100 Hz interpolation loop uses linear translation and rotation interpolation, then sends Cartesian targets through the RealMan pose-streaming interface; the robot controller resolves the joint motion. This is distinct from the lower-frequency diffusion replanning loop. The 100 Hz stream specifies the controller command schedule, which is separate from measured GPU-pipeline latency. The Pika SDK controls calibrated jaw opening. In Isaac Sim, the same TCP waypoints and gripper targets drive a Cartesian IK position controller with fixed joint limits and posture convention. Both domains use identical waypoint spacing, workspace bounds, and velocity/acceleration limits. Manufacturer interfaces and the RM75 model are documented publicly[1].

Frozen Feature Operators. The adapted encoders and restoration systems in Tab. A.3 are frozen before policy fitting. NoRA exports its complete NEA-L/14 operator; Theia and PriorDA retain their released ViT-B configurations.

Depth Restoration before Feature Extraction. CDM-L retains both RGB/depth encoders, cross-attention blocks, and the complete depth decoder. PromptDA-L retains its depth-prompt fusion and output layers. PriorDA-B/B retains coarse prediction, geometric alignment, and conditional fine prediction; LingBot-MDM-L retains its depth/mask heads and native postprocessing. Each complete system is adapted on both datasets before freezing and outputs restored depth for the common mixed-data-adapted DeFM-L/14 encoder, whose spatial features condition the policy. The restoration network and DeFM encoder both contribute to deployment cost.

Restored inverse depth is converted to metric depth where required; invalid output pixels are zeroed before DeFM preprocessing. All four cascades produce 1024-channel DeFM features for the common policy interface. The complete frozen cascade is used during policy training and deployment. Restoration methods receive RGB-D, whereas the representation panel receives depth; the protocol therefore evaluates complete pipelines with a shared policy architecture, without asserting identical input modalities.

Common Policy Interface. For a frozen spatial feature map with C_m channels, we concatenate its global spatial mean and a 4×4 adaptively pooled grid in a fixed order, yielding $17C_m$ values per observation. A trainable linear projection with LayerNorm maps this vector to 256 visual dimensions, which are concatenated with standardized proprioception. This projection is part of the policy, not the frozen representation. Its input width varies with C_m and its parameters are included in deployment cost; every subsequent policy layer has the same dimensions across methods. All encoders, restoration networks, normalization statistics, and NoRA heads/gate/basis remain frozen. Policies are fitted independently per method and task, with no reuse of the offline spatial or temporal predictors.

DP3 and Action Chunks. We adopt the conditional 1D U-Net DP3 (Ze et al., 2024), using the official low-dimensional global-conditioning implementation[2] with the learned visual projection above. The conditioning contains two historical observations; the prediction horizon is 16 actions, and eight actions beginning at index $T_o - 1$ are executed before replanning. The 10-dimensional Cartesian actions and 17-dimensional proprioception use the coordinate convention above; normalization is fitted only on simulated training data. The action waypoint rate is 10 Hz; the low-level robot controller tracks the same command representation in simulation and reality. All methods share the action dimension, observation history, execution horizon, and controller limits.

The U-Net has channel widths $[256, 512, 1024]$, a 256-dimensional diffusion-step embedding, kernel size 5, and eight GroupNorm groups. Training minimizes the standard noise-prediction MSE with 100 diffusion timesteps and a squared-cosine noise schedule. Inference uses 16 DDIM steps with $\eta = 0$ and one initial noise sample per replan; no best-of-N action selection is used. The projection and U-Net receive 100,000 AdamW updates, batch 128, learning rate 10^{-4}, $(\beta_1, \beta_2) = (0.95, 0.999)$, weight decay 10^{-6}, 500 warmup updates, cosine decay, and EMA following the official implementation.

Selection and Task Success. Seeds $\{0, 1, 2\}$ share demonstration indices and sampling schedules. The final-update EMA policy is fixed before evaluation, without selection on real trials. Simulation evaluation uses 100 held-out initial states per task and training seed. For each task/camera, each of the three policy seeds executes the same ten predeclared initial-state conditions, giving 30 trials in

[1] https://develop.realman-robotics.com/en/robot/json/motionConfig/, https://github.com/RealManRobot/rm_models, and https://github.com/agilexrobotics/pika_sdk.

[2] https://github.com/YanjieZe/3D-Diffusion-Policy.

total (10 per seed), as specified in the main table. Methods are paired by policy seed and initial-state ID; physical resets and method order follow the same randomized block schedule. The resulting budget is 450 real trials per method.

Game consists of Pick, Place, and Stack; Canteen consists of Pull, Fork, and Push; Kitchen consists of Open, Bread manipulation, and Close. One policy per task and training seed executes the full sequence. Initial objects, poses, and robot states are matched across methods, with method order randomized within each block. Each atomic column reports cumulative completion through that stage, using all 30 trials as denominator. Total requires all stages within 60 seconds, without manual intervention, and the terminal arrangement maintained for two seconds. Thus the last atomic stage and Total can differ when the final stability criterion fails.

D.4.2 Multi-Camera Generalization

The same frozen checkpoint and three policies per task are reused across all five cameras. Each trial uses one external camera, calibrated to a common workspace and crop, with matched mounting pose and fixed depth mode, frame rate and emitter/exposure settings. Methods share preprocessing for each camera; weights and feature statistics are not updated. Cameras operate separately to avoid sensor interference.

D435, L515, Azure Kinect, and ZED 2i occur in the declared mixed-data adaptation pool shared by all eight baselines and NoRA. All evaluated systems use mixed-data adaptation rather than camera-specific deployment updates. Gemini 336 is distinct from the Gemini 335/335L models listed in that pool; it is held out from adaptation in this protocol. Results therefore separate new scenes on seen camera models from transfer to the held-out Gemini 336 model. The held-out model belongs to a represented stereo-camera family, so this comparison tests camera-model replacement within the stated adaptation exposure. Unknown camera exposure in the original foundation-model pretraining is not asserted to be absent. No captures from the robot evaluation scenes or trials are used for adaptation or checkpoint selection.

Tab. 3 reports D435 full-task rates and an equal-weight mean over the three tasks. Other camera columns use the same task average; App. D.4.4 gives the cumulative-stage breakdown.

D.4.3 Deployment Efficiency

Measurements use trained checkpoints and real-camera inputs on a single NVIDIA GeForce RTX 4090 instance reporting 49,140 MiB of device memory, with driver 595.71.05, PyTorch 2.9.1+cu128, and CUDA 12.8. Inference uses batch size one and FP32, with TF32 disabled and no compilation, TensorRT, or quantization. Each method receives 50 warmup replans followed by 200 timed replans. Tab. 3 reports median latency; Tab. A.8 additionally reports p95.

Each replan uses one camera and two historical observations, with the previous observation's features cached. The steady-state path processes only the current 240×320 RGB/depth observation. DeFM, Theia, DINOv3, and NoRA use 224×224 encoder inputs; V-JEPA 2.1 uses its 384×384 image pathway. CDM and PromptDA restore at 518×518; PriorDA preserves aspect ratio at 518×686; LingBot-MDM uses a 30×40 RGB-token grid with native depth masking. Every restoration cascade then encodes the restored depth with DeFM-L at 224×224.

CUDA synchronization brackets host wall-clock timing, from decoded host inputs to a completed GPU action tensor. Timing includes host-to-device transfer, preprocessing, the complete feature pipeline, visual projection, proprioceptive conditioning, 16 DDIM steps, and 6D-rotation decoding. Camera exposure, file decoding, action transfer to the CPU/controller, network transport, and physical execution are excluded. Latency measures an entire action-chunk replan and is not divided by the eight executed actions. Startup with two uncached observations is outside this measurement.

Parameters cover all deployed feature modules, the visual projection, and one task policy. Restoration baselines include the full restoration network, its required output heads and postprocessing, and the separate DeFM-L encoder. The common diffusion U-Net has 73.05 M parameters; visual-projection sizes follow the feature-channel counts in App. D.4.1. Training-only teachers and Theia's feature translators are excluded.

The shared controller discards expired waypoints, interpolates remaining targets and holds the last target when the queue is empty. The same rules apply in simulation. Controller communication and physical tracking are excluded from GPU timing.

Table A.8: **Complete Feature-to-Action Deployment Cost.** *Latency* includes input preparation; *parameters* cover all executed modules.

Method	Params (M)	Median (ms)	p95 (ms)	Peak Alloc.(GiB)
PriorDA-B/14	575.97	175.6	180.4	2.453
PromptDA-L/14	721.21	148.3	153.1	3.134
CDM-L/14	1040.53	187.8	192.3	4.373
LingBot-MDM-L/14	702.33	192.6	197.5	2.973
Theia-B/16	162.19	97.4	103.1	0.633
DINOv3-L/16	380.66	108.3	114.1	1.445
V-JEPA 2.1-L/16	380.62	170.4	188.5	1.475
DeFM-L/14	381.16	101.7	108.4	1.468
NoRA-L/14	381.66	102.4	104.2	1.467

Memory measurement. Peak allocated GPU memory uses FP32, batch one and the full DP3 pipeline. The measurement window retains validation feature tensors through subsequent feature timing; it is not an isolated inference-memory measurement.

D.4.4 Statistics and Extended Reporting

For each task/camera, let k_s be complete-task successes out of ten trials for policy seed $s \in \{0, 1, 2\}$. Tables report mean $\pm$ sample standard deviation of the three seed-specific percentage rates (ddof $= 1$); the mean equals $100 \sum_s k_s/30$. Camera averages first average the three task rates per seed, then summarize across seeds. These standard deviations describe variability across the evaluated seed-specific policies and are not confidence intervals for a method difference.

The same trained policies are reused across cameras; the 450 trials per method are not independent trained-policy replicates. Reported seed deviations describe variability and are not confidence intervals for paired method differences.

Tabs. A.12–A.14 give cumulative stage and full-task success for each camera. Every entry uses the same trial denominator and seed-level aggregation.

E Ablation and Mechanism Analysis

Organization and Comparison References. Tables A.9–A.11 correspond to the three groups in Fig. 6. NIA is Ex.4 and NEA is Ex.20. The numerical scope is stated in App. D.

Shared Training and Evaluation. Configurations share Stage-1 EMA initialization, paired seeds $\{0, 1, 2\}$, scene splits, pair order, augmentations, correspondence masks and 30,000 Stage-2 updates, except where explicitly varied. Every changed operator receives fresh downstream heads and policies; Ex.18/Ex.19 instead branch from the Ex.20 checkpoint during policy fitting. Mechanism experiments use ViT-S/14, a ManiSkill3 geometry probe and D435 control, separately from the ViT-L native-width comparison.

Metrics and Capacity Controls. CGeo/NGeo are clean/noisy-input geometry errors from the same frozen-representation probe. We fit one tokenwise linear head per operator to ideal-depth patch mean and standard deviation on paired ManiSkill3 training frames, weighting clean/noisy views equally. Both test views share one head, the same valid-reference patches, metric units, and the same training-only target scaling across operators. CGeo/NGeo report patch-mean RMSE in mm. Errors are averaged within episodes, then equally across tasks, before averaging seeds. Neither teacher targets nor a model's gate define the geometry scoring support. This tests geometric information accessible to a fixed-capacity readout; patch means do not resolve all fine boundaries or contact details.

Sim/Real average full-task success equally over Game, Canteen and Kitchen. Simulation uses 100 matched initial states per task/seed; real evaluation uses ten states for each of three seeds, or 30 trials per task. Tables summarize the mean and sample standard deviation of seed-level task averages.

Native NIA/gate-only outputs have 1024 channels, whereas NEA/projection-only have 512. The main mechanism tables therefore use a common readout width: we apply the fixed isometric lift Φa to each rank-512 output token after the frozen operator and before the trainable readout; 1024-channel outputs use identity. The lift preserves rank and restores a common coordinate width. Geometry heads and the policy's 17×1024-to-256 projection then have identical trainable dimensions. Stage-2 objectives and operators are unchanged by this downstream capacity control. Equal-rank comparisons in the second group additionally hold the intrinsic bottleneck at 512.

E.1 Paired Alignment and Operator Components

Table A.9: **Paired alignment and operator components.** NoRA-NIA is Ex.4; NoRA-NEA is Ex.20. R1024 readouts control trainable input capacity. CGeo/NGeo: clean/noisy geometry RMSE (mm); Sim/Real: mean full-task success (%). Entries are mean ± sample std; bold denotes the best mean. Lower geometry error and higher success are better.

ID	Configuration	Pair	Gate	r	CGeo	NGeo	Sim	Real
Ex.1	Pure-Sim	–	–	1024	7.90 ± 0.19	23.69 ± 0.20	83.9 ± 2.0	33.3 ± 3.3
Ex.2	Clean continuation	clean	–	1024	**7.33** ± 0.06	24.43 ± 0.09	**87.1** ± 2.1	34.4 ± 3.8
Ex.3	Noise-Aug	synth.	–	1024	7.98 ± 0.09	15.34 ± 0.15	83.6 ± 1.9	60.0 ± 3.3
Ex.4	**NoRA-NIA**	real	–	1024	8.25 ± 0.08	12.61 ± 0.10	84.3 ± 1.2	71.1 ± 3.8
Ex.5	Gate only	real	yes	1024	8.77 ± 0.07	11.39 ± 0.10	83.8 ± 1.3	73.3 ± 3.3
Ex.6	Projection only	real	–	512	8.62 ± 0.07	11.10 ± 0.09	84.1 ± 1.6	75.6 ± 5.1
Ex.20	**NoRA-NEA**	real	yes	512	8.78 ± 0.06	**10.86** ± 0.11	84.6 ± 1.7	**77.8** ± 3.8

Ex.1/Ex.2 test extra optimization; Ex.3/Ex.4 test synthetic versus observed pairs. Ex.4/Ex.5/Ex.6/Ex.20 decompose the operator; these are distinct contrasts, not a single ranking against Ex.20.

Ex.1–Ex.6 and Ex.20: Training and Operator Factors. Ex.1 is Pure-Sim and omits Stage 2. Ex.2 applies the Stage-1 objective for another 30,000 updates to clean-reference sides of the same ordered training pairs. Ex.3 uses those references and their synthetic corruptions with the NIA objective, never their real-noisy partners. Ex.4 is real-pair NIA with $m = 1$, $\Phi = I$, and neither operator regularizer. Ex.5 learns the gate with $\mathcal{R}_{\text{gate}}$ but fixes $\Phi = I$ and omits $\mathcal{R}_{\text{proj}}$. Ex.6 fixes $m = 1$, omits $\mathcal{R}_{\text{gate}}$, and learns rank-512 Φ with $\mathcal{R}_{\text{proj}}$. Ex.20 learns both. Ex.3–Ex.6/Ex.20 retain the same clean anchor, deployment term, directional weights, and remaining coefficients. Equal updates do not equalize crop counts or teacher computation for clean continuation.

Ex.4/Ex.5/Ex.6/Ex.20 compare gate and projection components under matched readout capacity; changing projection also changes intrinsic rank. Clean/noisy geometry and control jointly characterize this trade-off.

For Ex.3, we use the following synthetic corruptions. At severity $j \in \{0, 1, 2\}$, missing rectangles remove fraction $0.05j$ of originally valid pixels (side lengths 4–24); flying pixels replace fraction $0.05j$ of a two-pixel reference-depth boundary band by uniform foreground/background interpolation; range error adds Gaussian noise of standard deviation $0.0025jD^2/d_{\max}$ and a scene-wise signed bias $\pm0.0025jD$. We fix $d_{\max}$ on source training and sample the three severities uniformly during representation adaptation. All corresponding policies still use clean simulation observations; Ex.3 isolates synthetic versus observed alignment pairs, not policy-training noise augmentation.

E.2 Stability Supervision and Geometry Retention

Ex.7–Ex.9: Source of Gate Supervision. Ex.7 permutes the teacher top-ρ binary target independently within each pair's eligible spatial locations, using the same permutation for its two views; the exact positive count is preserved. Ex.8 constructs scores from the fraction of valid raw noisy-depth pixels in each patch and takes the same top-$\lfloor \rho N \rfloor$ count. Ties are broken with a seeded, location-independent random ordering shared across views. Ex.9 removes only gate BCE, retaining agreement, budget, entropy, and all alignment/interface gradients. Gate architecture, projector supervision, and remaining coefficients match Ex.20. Equal target counts do not guarantee equal learned masses. These

Table A.10: **Stability supervision and geometric information.** All rows retain rank 512 and the same gate architecture. Disp./Keep denote the two optimized projection terms. The learned-bottleneck control (Ex.10) retains basis gradients from all other objectives. Metrics and R1024 readouts follow Tab. A.9.

ID	Gate target	Basis	Disp.	Keep	CGeo	NGeo	Real
Ex.7	Shuffled stability	learned	yes	yes	9.75 ± 0.10	13.14 ± 0.17	64.4 ± 5.1
Ex.8	Raw validity	learned	yes	yes	9.19 ± 0.13	11.96 ± 0.09	68.9 ± 1.9
Ex.9	No BCE	learned	yes	yes	9.00 ± 0.06	11.21 ± 0.14	73.3 ± 0.0
Ex.10	Teacher stability	learned	–	–	8.64 ± 0.06	11.33 ± 0.08	74.4 ± 1.9
Ex.11	Teacher stability	random	–	–	9.75 ± 0.12	12.84 ± 0.21	66.7 ± 3.3
Ex.12	Teacher stability	PCA	–	–	**8.27** ± 0.07	11.99 ± 0.15	71.1 ± 3.8
Ex.13	Teacher stability	learned	–	yes	8.59 ± 0.07	11.66 ± 0.14	73.3 ± 0.0
Ex.14	Teacher stability	learned	yes	–	10.30 ± 0.19	11.92 ± 0.20	68.9 ± 6.9
Ex.20	**Teacher stability**	**learned**	yes	yes	8.78 ± 0.06	**10.86** ± 0.11	**77.8** ± 3.8

Ex.7–Ex.9 versus NEA (Ex.20) test gate supervision. Ex.10–Ex.12 versus Ex.20 separate learned compression from stability guidance; Ex.13/Ex.14 isolate the two projection terms. These aggregate metrics assess geometric accessibility and control, not region-level retention.

controls compare teacher stability with generic masking, missing-depth avoidance, and unconstrained learned attention.

Ex.10–Ex.14: Stability Guidance versus Compression. All five variants retain Ex.20's gate and rank 512. Ex.10 sets the outer $\lambda_{\text{proj}} = 0$ but keeps Φ trainable through matching and deployment losses and retracts it by QR. Ex.11 uses a fixed seeded Gaussian-QR basis. Ex.12 uses the fixed top-512 PCA directions of normalized clean teacher patch tokens from source-training scenes only. PCA uses centering to estimate directions; the resulting basis uses the unchanged NoRA interface, without adding a centering transform at deployment. Fixed bases in Ex.11/Ex.12 have no optimized projection regularizer. Ex.13 deletes only $(1 - s_i)\|Q_\Phi \Delta_i^t\|^2$ from $\mathcal{R}_{\text{proj}}$; Ex.14 sets $\lambda_{\text{keep}} = 0$ and retains displacement suppression. All other objective terms remain active. Ex.10 tests learned compression without stability regularization; Ex.11/Ex.12 hold rank fixed with independently constructed bases.

E.3 Geometry Anchoring and the Policy Interface

Table A.11: **Geometry anchoring and policy-interface preservation.** Ex.15–Ex.17 retrain Stage 2; Ex.18/Ex.19 branch from the same NEA (Ex.20) checkpoint only at policy training. Anchor and Deploy denote active Stage-2 losses. All rows keep the 256-dimensional policy projection trainable. Metrics and formatting follow Tab. A.9.

ID	Anchor	α	Deploy	Perception tuning	CGeo	NGeo	Sim	Real
Ex.15	–	2/3	yes	none	10.77 ± 0.73	11.62 ± 0.92	79.3 ± 2.3	71.1 ± 5.1
Ex.16	yes	1	yes	none	8.75 ± 0.06	**10.55** ± 0.08	85.0 ± 1.2	**78.9** ± 1.9
Ex.17	yes	2/3	–	none	**8.73** ± 0.09	11.18 ± 0.11	84.0 ± 1.2	70.0 ± 5.8
Ex.18	yes	2/3	yes	heads, gate, basis	9.16 ± 0.14	12.45 ± 0.22	89.0 ± 1.5	73.3 ± 3.3
Ex.19	yes	2/3	yes	entire operator	10.20 ± 0.31	15.54 ± 0.50	**92.2** ± 0.8	70.0 ± 3.3
Ex.20	yes	2/3	yes	**none (NEA)**	8.78 ± 0.06	10.86 ± 0.11	84.6 ± 1.7	77.8 ± 3.8

Geometry entries for Ex.18/Ex.19 use the post-policy operator for each task and newly fitted diagnostic probes. These errors assess accessible geometry after policy fitting.

Ex.15–Ex.17: Preservation and Direction. Ex.15 removes only the unprojected clean-view anchor ($\lambda_{\text{sem}} = 0$), retaining both projection terms. Ex.16 sets $\alpha = 1$ in both scene and token matching; their outer coefficients remain two, so the total directional weight is unchanged. Ex.17 removes only $\mathcal{L}_{\text{deploy}}$ ($\lambda_{\text{dep}} = 0$), retaining gate supervision and selective teacher matching. Ex.16 has slightly better means than the default; $\alpha = 2/3$ remains the common configuration for the other comparisons.

Exact Interface Boundary. The robot interface is the $17C_m$-dimensional pooled representation in App. D.4.1. Its trainable 256-dimensional policy projection is outside the perception operator and consistency loss. The fixed R1024 lift controls downstream capacity without changing Stage-2 training.

Ex.18/Ex.19: Freezing during Policy Learning. Ex.20 freezes the entire operator. Ex.18 freezes only the backbone and updates heads, gate, and basis; Ex.19 additionally updates the backbone. All three start from the same Stage-2 Ex.20 checkpoint, use simulated action supervision only, retain the same policy learning rate (10^{-4}), update budget, demonstrations, and policy initialization, and use QR whenever Φ is updated. For each unfrozen alternative, perception learning rate is selected from $\{3\times10^{-6}, 10^{-5}, 3\times10^{-5}\}$ with an equal simulation-validation budget, before real testing. The policy's 256-dimensional projection remains trainable in all variants. In R1024, the fixed lift uses the pre-policy basis and remains fixed even when the internal basis is fine-tuned. These controls test whether simulated policy fitting alters the aligned perception interface.

F LIMITATIONS

Reference Supervision and Geometric Trade-offs. NoRA relies on corresponding reference/noisy observations and a pretrained depth teacher. LingBot-Depth pairs and Depth Anything 3-derived ByteCameraDepth references provide alignment supervision; model-generated references are not independent physical ground truth. Teacher disagreement combines sensing variation with reference and registration errors. Patch-depth averages do not resolve every thin structure or contact detail; the low-rank trade-off is evaluated only at the stated rank and retention ratio.

Sensing and Task Coverage. The real-robot evaluation concerns indoor tabletop tasks on one arm–gripper platform. Four tested camera models occur in adaptation; Gemini 336 is held out while related Gemini models are seen. These results support camera replacement in the evaluated setting, rather than universal transfer to unseen sensor families. Outdoor sunlight, broader task distributions, mobile manipulation, and bimanual systems remain outside the real-control evaluation. Zero-shot transfer refers to the absence of real-robot policy training after real-depth encoder alignment.

Predictive Evaluation Scope. Spatial completion receives full RGB context and temporal forecasting receives observed depth history without future actions. Shared DeFM-target cosine similarity measures predictive accessibility in that representation space. It does not alone establish depth-only geometric recovery, an advantage specific to moving regions, or action-conditioned causal dynamics. The robot comparison fixes clean simulation policy training; it does not evaluate the interaction between selective alignment and strong policy-training noise augmentation.

G Extended Robot Results and Visualizations

Table A.12: **Cross-Camera Sim-to-Real Manipulation: Game.** Cumulative stage and complete-task success (%, ↑), mean ± sample std over 3 seeds with 10 trials each.

Camera	Method	Pick	Place	Stack	Total
RealSense D435	DeFM-L/14	**96.7** ± 5.8	90.0 ± 10.0	83.3 ± 5.8	80.0 ± 10.0
	Theia-B/16	93.3 ± 5.8	83.3 ± 5.8	70.0 ± 10.0	70.0 ± 10.0
	DINOv3-L/16	**96.7** ± 5.8	90.0 ± 0.0	76.7 ± 5.8	76.7 ± 5.8
	V-JEPA 2.1-L/16	93.3 ± 5.8	86.7 ± 5.8	76.7 ± 5.8	76.7 ± 5.8
	CDM-L/14	**96.7** ± 5.8	**93.3** ± 5.8	**86.7** ± 5.8	83.3 ± 5.8
	PromptDA-L/14	93.3 ± 5.8	86.7 ± 5.8	73.3 ± 11.6	73.3 ± 11.6
	PriorDA-B/14	93.3 ± 5.8	83.3 ± 5.8	70.0 ± 10.0	70.0 ± 10.0
	LingBot-MDM-L/14	**96.7** ± 5.8	90.0 ± 10.0	83.3 ± 11.6	80.0 ± 10.0
	NoRA-L/14	**96.7** ± 5.8	**93.3** ± 5.8	**86.7** ± 5.8	**86.7** ± 5.8
RealSense L515	DeFM-L/14	93.3 ± 5.8	86.7 ± 11.6	76.7 ± 5.8	76.7 ± 5.8
	Theia-B/16	93.3 ± 5.8	80.0 ± 0.0	66.7 ± 5.8	66.7 ± 5.8
	DINOv3-L/16	**96.7** ± 5.8	90.0 ± 0.0	76.7 ± 5.8	76.7 ± 5.8
	V-JEPA 2.1-L/16	**96.7** ± 5.8	86.7 ± 5.8	73.3 ± 5.8	73.3 ± 5.8
	CDM-L/14	**96.7** ± 5.8	90.0 ± 0.0	83.3 ± 5.8	80.0 ± 10.0
	PromptDA-L/14	93.3 ± 5.8	83.3 ± 5.8	70.0 ± 10.0	70.0 ± 10.0
	PriorDA-B/14	93.3 ± 5.8	80.0 ± 0.0	66.7 ± 5.8	66.7 ± 5.8
	LingBot-MDM-L/14	**96.7** ± 5.8	**93.3** ± 5.8	**86.7** ± 5.8	83.3 ± 5.8
	NoRA-L/14	**96.7** ± 5.8	**93.3** ± 5.8	**86.7** ± 5.8	**86.7** ± 5.8
Orbbec Gemini 336	DeFM-L/14	93.3 ± 5.8	86.7 ± 5.8	80.0 ± 10.0	80.0 ± 10.0
	Theia-B/16	93.3 ± 5.8	83.3 ± 5.8	70.0 ± 10.0	70.0 ± 10.0
	DINOv3-L/16	**96.7** ± 5.8	86.7 ± 5.8	73.3 ± 5.8	73.3 ± 5.8
	V-JEPA 2.1-L/16	93.3 ± 5.8	86.7 ± 5.8	76.7 ± 5.8	76.7 ± 5.8
	CDM-L/14	93.3 ± 5.8	86.7 ± 5.8	80.0 ± 0.0	80.0 ± 0.0
	PromptDA-L/14	**96.7** ± 5.8	86.7 ± 5.8	73.3 ± 5.8	73.3 ± 5.8
	PriorDA-B/14	93.3 ± 5.8	80.0 ± 0.0	70.0 ± 0.0	70.0 ± 0.0
	LingBot-MDM-L/14	**96.7** ± 5.8	**93.3** ± 5.8	**86.7** ± 5.8	83.3 ± 5.8
	NoRA-L/14	**96.7** ± 5.8	90.0 ± 10.0	**86.7** ± 5.8	**86.7** ± 5.8
Stereolabs ZED 2i	DeFM-L/14	93.3 ± 5.8	83.3 ± 5.8	76.7 ± 11.6	73.3 ± 5.8
	Theia-B/16	86.7 ± 5.8	76.7 ± 5.8	63.3 ± 5.8	63.3 ± 5.8
	DINOv3-L/16	93.3 ± 5.8	80.0 ± 0.0	66.7 ± 5.8	66.7 ± 5.8
	V-JEPA 2.1-L/16	**96.7** ± 5.8	86.7 ± 5.8	73.3 ± 5.8	73.3 ± 5.8
	CDM-L/14	93.3 ± 5.8	86.7 ± 5.8	76.7 ± 5.8	76.7 ± 5.8
	PromptDA-L/14	93.3 ± 5.8	80.0 ± 0.0	66.7 ± 5.8	66.7 ± 5.8
	PriorDA-B/14	86.7 ± 5.8	76.7 ± 5.8	63.3 ± 5.8	63.3 ± 5.8
	LingBot-MDM-L/14	93.3 ± 5.8	**90.0** ± 10.0	**83.3** ± 5.8	80.0 ± 10.0
	NoRA-L/14	**96.7** ± 5.8	**90.0** ± 0.0	**83.3** ± 5.8	**83.3** ± 5.8
Azure Kinect	DeFM-L/14	93.3 ± 5.8	90.0 ± 0.0	83.3 ± 5.8	80.0 ± 10.0
	Theia-B/16	93.3 ± 5.8	80.0 ± 0.0	66.7 ± 5.8	66.7 ± 5.8
	DINOv3-L/16	93.3 ± 5.8	86.7 ± 5.8	73.3 ± 5.8	73.3 ± 5.8
	V-JEPA 2.1-L/16	93.3 ± 5.8	86.7 ± 5.8	76.7 ± 5.8	76.7 ± 5.8
	CDM-L/14	93.3 ± 5.8	86.7 ± 5.8	83.3 ± 5.8	80.0 ± 10.0
	PromptDA-L/14	**96.7** ± 5.8	83.3 ± 5.8	70.0 ± 0.0	70.0 ± 0.0
	PriorDA-B/14	93.3 ± 5.8	83.3 ± 5.8	70.0 ± 10.0	70.0 ± 10.0
	LingBot-MDM-L/14	93.3 ± 5.8	90.0 ± 0.0	83.3 ± 5.8	80.0 ± 0.0
	NoRA-L/14	**96.7** ± 5.8	**93.3** ± 5.8	**86.7** ± 5.8	**86.7** ± 5.8

Table A.13: **Cross-Camera Sim-to-Real Manipulation: Canteen.** Cumulative stage and complete-task success (%, ↑), mean ± sample std over 3 seeds with 10 trials each.

Camera	Method	Pull	Fork	Push	Total
RealSense D435	DeFM-L/14	90.0 ± 10.0	73.3 ± 11.6	70.0 ± 10.0	66.7 ± 15.3
	Theia-B/16	86.7 ± 5.8	70.0 ± 10.0	60.0 ± 10.0	60.0 ± 10.0
	DINOv3-L/16	86.7 ± 5.8	66.7 ± 15.3	56.7 ± 15.3	56.7 ± 15.3
	V-JEPA 2.1-L/16	86.7 ± 5.8	73.3 ± 5.8	66.7 ± 11.6	66.7 ± 11.6
	CDM-L/14	**93.3** ± 5.8	**80.0** ± 0.0	76.7 ± 5.8	73.3 ± 5.8
	PromptDA-L/14	86.7 ± 5.8	66.7 ± 5.8	56.7 ± 5.8	56.7 ± 5.8
	PriorDA-B/14	90.0 ± 0.0	76.7 ± 5.8	66.7 ± 5.8	63.3 ± 5.8
	LingBot-MDM-L/14	**93.3** ± 5.8	**80.0** ± 10.0	**80.0** ± 10.0	**80.0** ± 10.0
	NoRA-L/14	**93.3** ± 5.8	**80.0** ± 10.0	76.7 ± 5.8	76.7 ± 5.8
RealSense L515	DeFM-L/14	90.0 ± 10.0	70.0 ± 10.0	66.7 ± 11.6	63.3 ± 15.3
	Theia-B/16	83.3 ± 5.8	63.3 ± 5.8	53.3 ± 5.8	53.3 ± 5.8
	DINOv3-L/16	86.7 ± 5.8	70.0 ± 10.0	60.0 ± 10.0	60.0 ± 10.0
	V-JEPA 2.1-L/16	86.7 ± 5.8	70.0 ± 0.0	63.3 ± 5.8	63.3 ± 5.8
	CDM-L/14	**93.3** ± 5.8	**80.0** ± 10.0	76.7 ± 5.8	73.3 ± 5.8
	PromptDA-L/14	86.7 ± 5.8	66.7 ± 5.8	56.7 ± 5.8	56.7 ± 5.8
	PriorDA-B/14	86.7 ± 5.8	66.7 ± 5.8	56.7 ± 5.8	56.7 ± 5.8
	LingBot-MDM-L/14	**93.3** ± 5.8	**80.0** ± 10.0	**80.0** ± 10.0	**80.0** ± 10.0
	NoRA-L/14	**93.3** ± 5.8	**80.0** ± 0.0	76.7 ± 5.8	76.7 ± 5.8
Orbbec Gemini 336	DeFM-L/14	86.7 ± 5.8	70.0 ± 10.0	66.7 ± 15.3	63.3 ± 15.3
	Theia-B/16	86.7 ± 5.8	70.0 ± 10.0	60.0 ± 10.0	60.0 ± 10.0
	DINOv3-L/16	86.7 ± 5.8	73.3 ± 5.8	63.3 ± 5.8	63.3 ± 5.8
	V-JEPA 2.1-L/16	90.0 ± 0.0	73.3 ± 5.8	66.7 ± 5.8	66.7 ± 5.8
	CDM-L/14	90.0 ± 0.0	73.3 ± 5.8	70.0 ± 0.0	70.0 ± 0.0
	PromptDA-L/14	90.0 ± 0.0	70.0 ± 10.0	60.0 ± 10.0	60.0 ± 10.0
	PriorDA-B/14	86.7 ± 5.8	70.0 ± 0.0	60.0 ± 0.0	60.0 ± 0.0
	LingBot-MDM-L/14	**93.3** ± 5.8	**80.0** ± 10.0	**80.0** ± 10.0	**80.0** ± 10.0
	NoRA-L/14	**93.3** ± 5.8	76.7 ± 5.8	76.7 ± 5.8	76.7 ± 5.8
Stereolabs ZED 2i	DeFM-L/14	86.7 ± 5.8	70.0 ± 10.0	63.3 ± 15.3	63.3 ± 15.3
	Theia-B/16	83.3 ± 5.8	56.7 ± 5.8	46.7 ± 5.8	46.7 ± 5.8
	DINOv3-L/16	83.3 ± 5.8	60.0 ± 10.0	50.0 ± 10.0	50.0 ± 10.0
	V-JEPA 2.1-L/16	86.7 ± 5.8	70.0 ± 10.0	60.0 ± 10.0	60.0 ± 10.0
	CDM-L/14	**90.0** ± 0.0	73.3 ± 5.8	70.0 ± 10.0	66.7 ± 5.8
	PromptDA-L/14	80.0 ± 0.0	53.3 ± 5.8	43.3 ± 5.8	43.3 ± 5.8
	PriorDA-B/14	83.3 ± 5.8	56.7 ± 5.8	46.7 ± 5.8	46.7 ± 5.8
	LingBot-MDM-L/14	**90.0** ± 0.0	**76.7** ± 5.8	**73.3** ± 5.8	**70.0** ± 0.0
	NoRA-L/14	**90.0** ± 0.0	**76.7** ± 5.8	**73.3** ± 5.8	**70.0** ± 0.0
Azure Kinect	DeFM-L/14	90.0 ± 10.0	73.3 ± 5.8	66.7 ± 5.8	63.3 ± 11.6
	Theia-B/16	83.3 ± 5.8	66.7 ± 5.8	56.7 ± 5.8	56.7 ± 5.8
	DINOv3-L/16	86.7 ± 5.8	70.0 ± 10.0	60.0 ± 10.0	60.0 ± 10.0
	V-JEPA 2.1-L/16	86.7 ± 5.8	73.3 ± 5.8	63.3 ± 5.8	63.3 ± 5.8
	CDM-L/14	**93.3** ± 5.8	**80.0** ± 10.0	**76.7** ± 5.8	73.3 ± 5.8
	PromptDA-L/14	86.7 ± 5.8	66.7 ± 5.8	56.7 ± 5.8	56.7 ± 5.8
	PriorDA-B/14	86.7 ± 5.8	70.0 ± 0.0	60.0 ± 0.0	60.0 ± 0.0
	LingBot-MDM-L/14	**93.3** ± 5.8	**80.0** ± 10.0	**76.7** ± 5.8	**76.7** ± 5.8
	NoRA-L/14	**93.3** ± 5.8	**80.0** ± 0.0	**76.7** ± 5.8	**76.7** ± 5.8

Table A.14: **Cross-Camera Sim-to-Real Manipulation: Kitchen.** Cumulative stage and complete-task success (%, ↑), mean ± sample std over 3 seeds with 10 trials each.

Camera	Method	Open	Bread	Close	Total
RealSense D435	DeFM-L/14	90.0 ± 10.0	80.0 ± 0.0	70.0 ± 0.0	66.7 ± 5.8
	Theia-B/16	83.3 ± 5.8	66.7 ± 5.8	53.3 ± 11.6	53.3 ± 11.6
	DINOv3-L/16	86.7 ± 5.8	73.3 ± 5.8	60.0 ± 10.0	56.7 ± 5.8
	V-JEPA 2.1-L/16	86.7 ± 5.8	73.3 ± 5.8	66.7 ± 5.8	63.3 ± 5.8
	CDM-L/14	**93.3** ± 5.8	83.3 ± 11.6	73.3 ± 11.6	73.3 ± 11.6
	PromptDA-L/14	83.3 ± 5.8	66.7 ± 5.8	53.3 ± 11.6	50.0 ± 10.0
	PriorDA-B/14	83.3 ± 5.8	70.0 ± 0.0	56.7 ± 5.8	53.3 ± 5.8
	LingBot-MDM-L/14	**93.3** ± 5.8	**86.7** ± 5.8	80.0 ± 10.0	76.7 ± 5.8
	NoRA-L/14	**93.3** ± 5.8	**86.7** ± 5.8	**83.3** ± 5.8	**80.0** ± 10.0
RealSense L515	DeFM-L/14	90.0 ± 0.0	76.7 ± 5.8	66.7 ± 5.8	63.3 ± 11.6
	Theia-B/16	83.3 ± 5.8	66.7 ± 5.8	53.3 ± 11.6	53.3 ± 11.6
	DINOv3-L/16	86.7 ± 5.8	73.3 ± 5.8	60.0 ± 10.0	56.7 ± 5.8
	V-JEPA 2.1-L/16	86.7 ± 5.8	73.3 ± 5.8	63.3 ± 5.8	60.0 ± 0.0
	CDM-L/14	**93.3** ± 5.8	**83.3** ± 5.8	76.7 ± 5.8	73.3 ± 5.8
	PromptDA-L/14	86.7 ± 5.8	70.0 ± 10.0	56.7 ± 5.8	56.7 ± 5.8
	PriorDA-B/14	83.3 ± 5.8	70.0 ± 0.0	56.7 ± 5.8	53.3 ± 5.8
	LingBot-MDM-L/14	**93.3** ± 5.8	**83.3** ± 5.8	76.7 ± 5.8	73.3 ± 5.8
	NoRA-L/14	**93.3** ± 5.8	**83.3** ± 5.8	**80.0** ± 0.0	**76.7** ± 5.8
Orbbec Gemini 336	DeFM-L/14	90.0 ± 0.0	80.0 ± 10.0	70.0 ± 10.0	66.7 ± 15.3
	Theia-B/16	83.3 ± 5.8	66.7 ± 5.8	56.7 ± 5.8	53.3 ± 11.6
	DINOv3-L/16	86.7 ± 5.8	73.3 ± 5.8	60.0 ± 0.0	60.0 ± 0.0
	V-JEPA 2.1-L/16	86.7 ± 5.8	76.7 ± 5.8	66.7 ± 5.8	63.3 ± 5.8
	CDM-L/14	90.0 ± 0.0	80.0 ± 0.0	73.3 ± 5.8	70.0 ± 0.0
	PromptDA-L/14	83.3 ± 5.8	66.7 ± 5.8	56.7 ± 5.8	53.3 ± 5.8
	PriorDA-B/14	83.3 ± 5.8	73.3 ± 5.8	60.0 ± 10.0	56.7 ± 11.6
	LingBot-MDM-L/14	**93.3** ± 5.8	83.3 ± 5.8	80.0 ± 0.0	76.7 ± 5.8
	NoRA-L/14	**93.3** ± 5.8	**86.7** ± 5.8	**83.3** ± 5.8	**80.0** ± 10.0
Stereolabs ZED 2i	DeFM-L/14	86.7 ± 5.8	73.3 ± 11.6	63.3 ± 11.6	60.0 ± 10.0
	Theia-B/16	80.0 ± 0.0	63.3 ± 5.8	46.7 ± 5.8	46.7 ± 5.8
	DINOv3-L/16	83.3 ± 5.8	66.7 ± 5.8	53.3 ± 5.8	53.3 ± 5.8
	V-JEPA 2.1-L/16	86.7 ± 5.8	73.3 ± 5.8	60.0 ± 10.0	56.7 ± 5.8
	CDM-L/14	90.0 ± 0.0	76.7 ± 5.8	70.0 ± 10.0	66.7 ± 5.8
	PromptDA-L/14	80.0 ± 0.0	63.3 ± 5.8	46.7 ± 5.8	46.7 ± 5.8
	PriorDA-B/14	80.0 ± 0.0	63.3 ± 5.8	46.7 ± 5.8	46.7 ± 5.8
	LingBot-MDM-L/14	**93.3** ± 5.8	83.3 ± 5.8	76.7 ± 5.8	73.3 ± 5.8
	NoRA-L/14	**93.3** ± 5.8	**86.7** ± 5.8	**80.0** ± 0.0	**76.7** ± 5.8
Azure Kinect	DeFM-L/14	90.0 ± 10.0	80.0 ± 17.3	70.0 ± 17.3	66.7 ± 20.8
	Theia-B/16	83.3 ± 5.8	66.7 ± 5.8	53.3 ± 11.6	53.3 ± 11.6
	DINOv3-L/16	86.7 ± 5.8	73.3 ± 5.8	63.3 ± 5.8	60.0 ± 0.0
	V-JEPA 2.1-L/16	86.7 ± 5.8	76.7 ± 5.8	66.7 ± 5.8	63.3 ± 5.8
	CDM-L/14	90.0 ± 0.0	83.3 ± 5.8	76.7 ± 5.8	73.3 ± 5.8
	PromptDA-L/14	83.3 ± 5.8	66.7 ± 5.8	56.7 ± 5.8	56.7 ± 5.8
	PriorDA-B/14	83.3 ± 5.8	70.0 ± 0.0	56.7 ± 5.8	53.3 ± 5.8
	LingBot-MDM-L/14	**93.3** ± 5.8	**86.7** ± 5.8	80.0 ± 0.0	76.7 ± 5.8
	NoRA-L/14	**93.3** ± 5.8	**86.7** ± 5.8	**83.3** ± 5.8	**80.0** ± 10.0

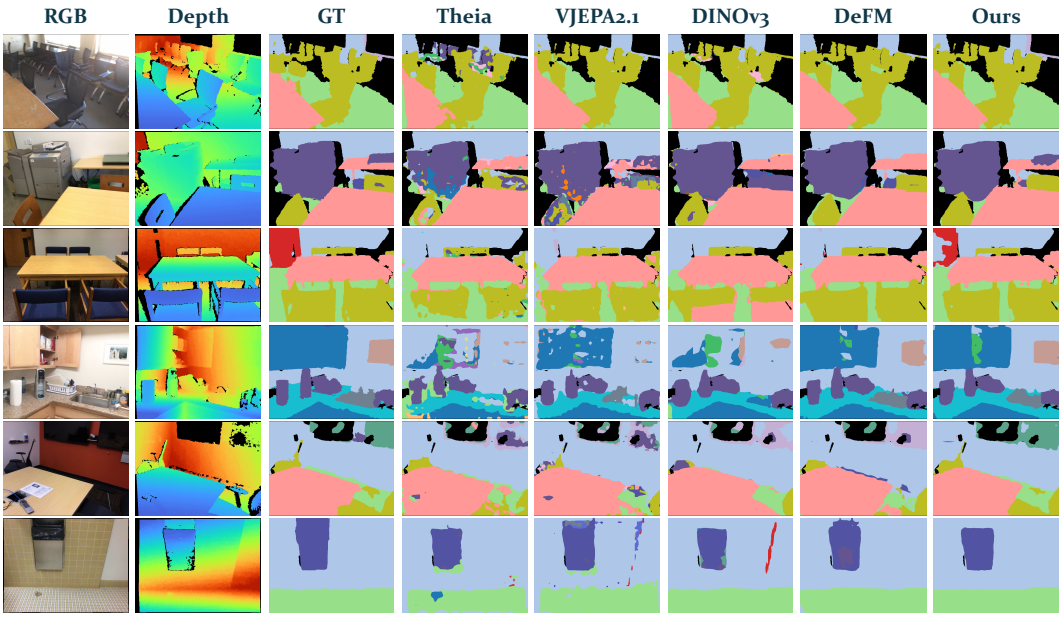

Figure A.1: **Additional Segmentation Results on ScanNet.** Qualitative comparisons on indoor ScanNet scenes. Columns show RGB, depth, ground truth, Theia, V-JEPA 2.1, DINOv3, DeFM, and NoRA.

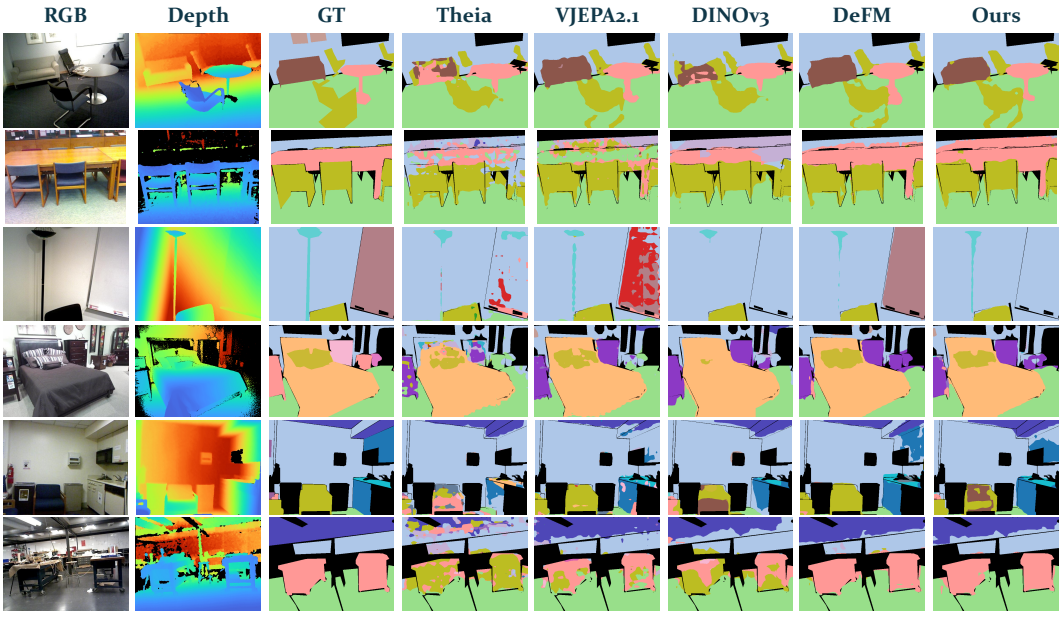

Figure A.2: **Additional Segmentation Results on SUN-RGBD.** Qualitative comparisons on indoor SUN-RGBD scenes. Columns show RGB, depth, ground truth, Theia, V-JEPA 2.1, DINOv3, DeFM, and NoRA.

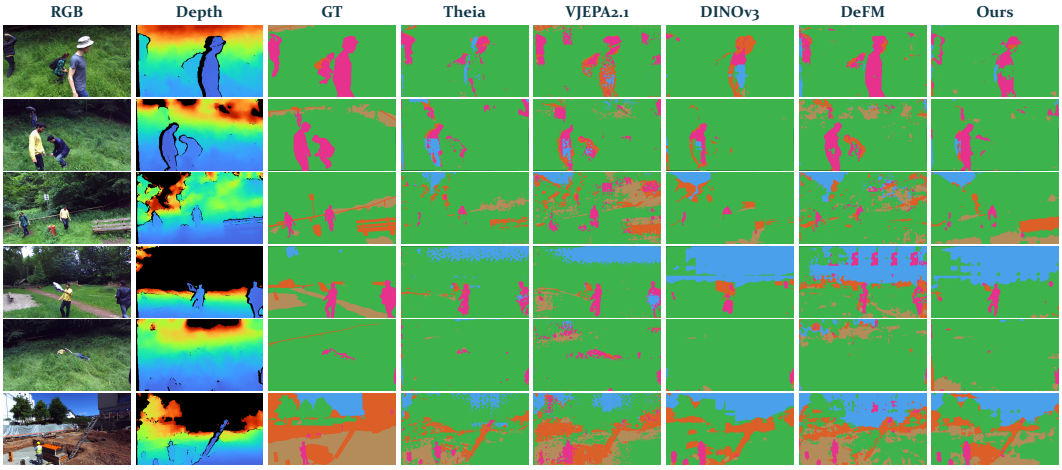

Figure A.3: **Additional Segmentation Results on OFFSED.** Qualitative comparisons on outdoor OFFSED scenes. Columns show RGB, depth, ground truth, Theia, V-JEPA 2.1, DINOv3, DeFM, and NoRA.

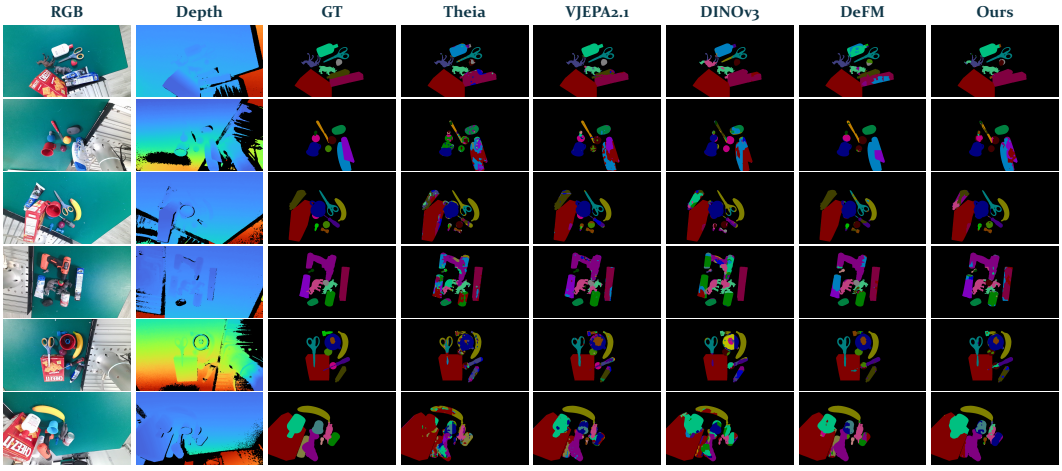

Figure A.4: **Additional Segmentation Results on GraspNet-1B.** Qualitative comparisons on GraspNet-1B tabletop scenes. Columns show RGB, depth, ground truth, Theia, V-JEPA 2.1, DINOv3, DeFM, and NoRA.

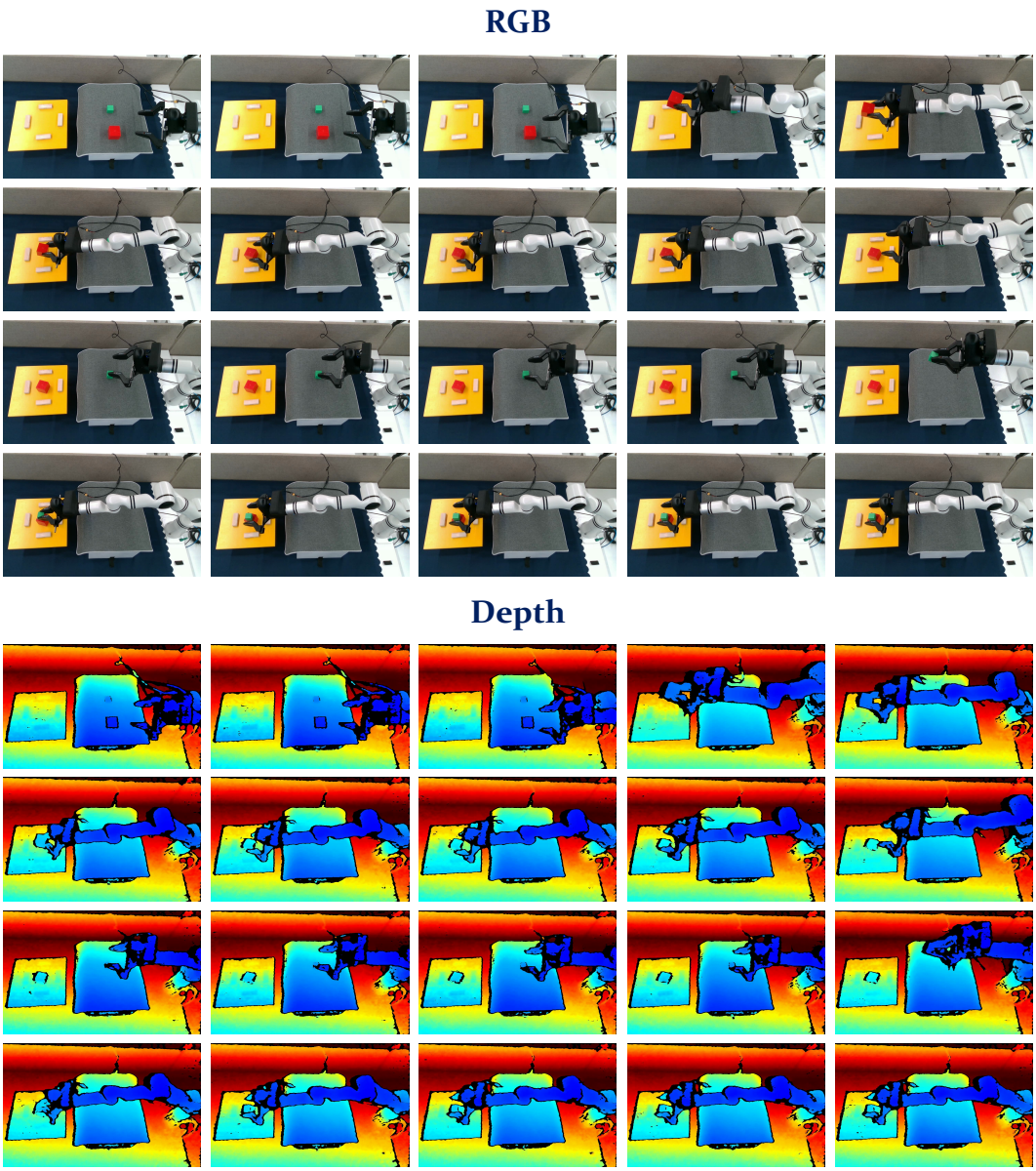

Figure A.5: **Additional Game Task Rollout.** RGB observations (upper block) and corresponding depth observations (lower block) show the sequence of picking, placing, and stacking. Within each block, frames progress from left to right and then from top to bottom.

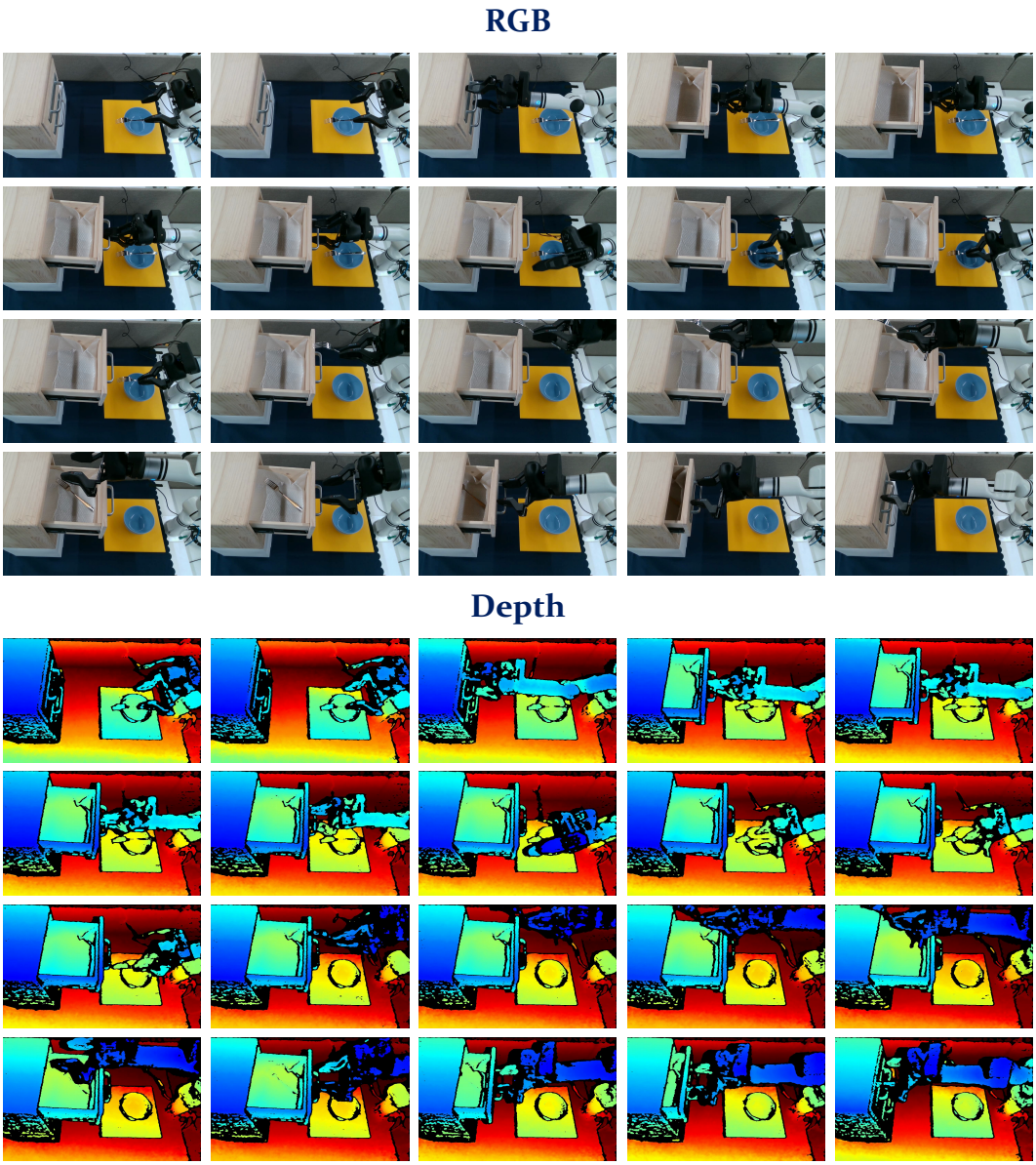

Figure A.6: **Additional Canteen Task Rollout.** RGB observations (upper block) and corresponding depth observations (lower block) show the sequence of pulling, forking, and pushing. Within each block, frames progress from left to right and then from top to bottom.

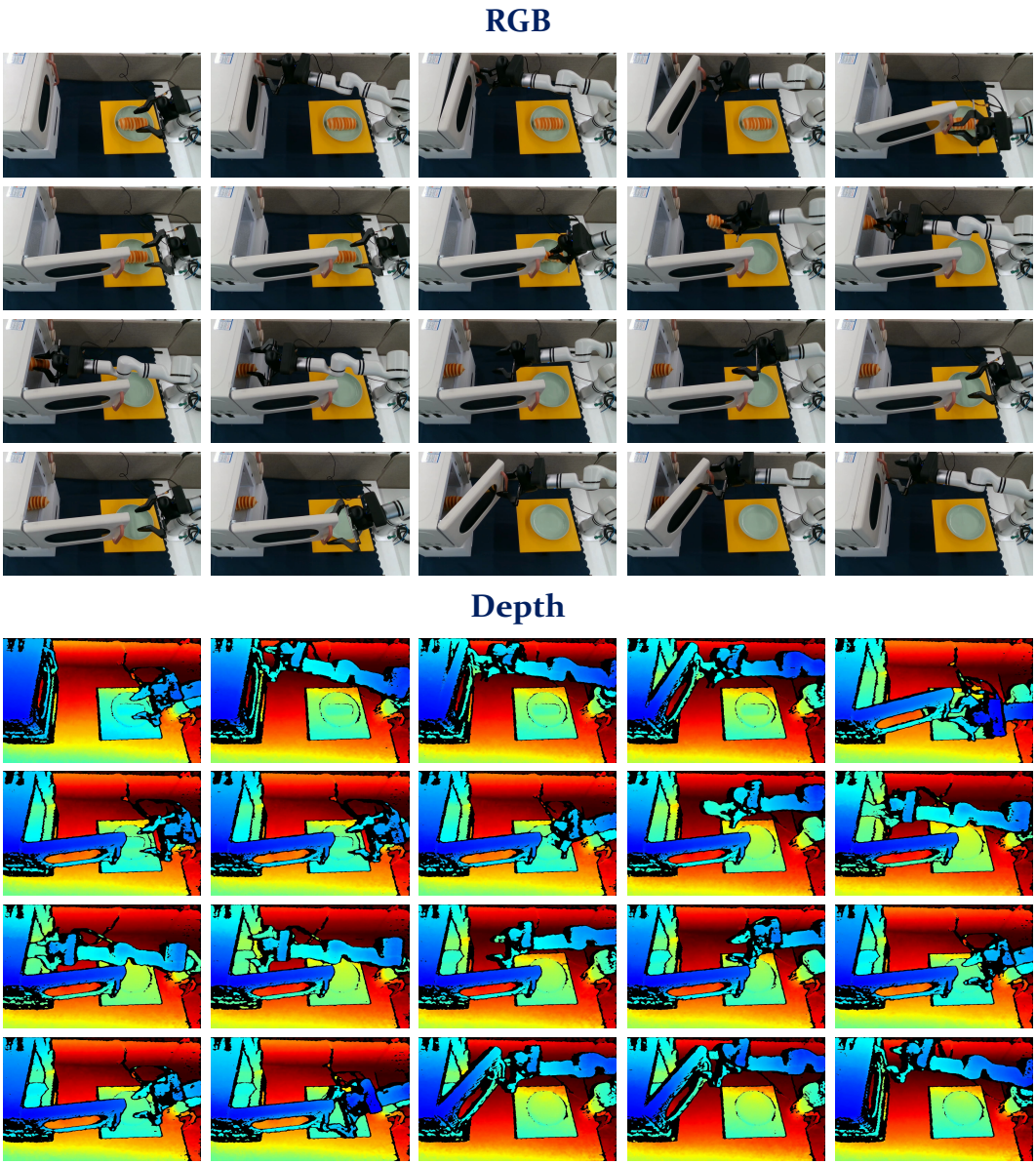

Figure A.7: **Additional Kitchen Task Rollout.** RGB observations (upper block) and corresponding depth observations (lower block) show the sequence of opening, bread manipulation, and closing. Within each block, frames progress from left to right and then from top to bottom.