FOVI-Dex: Foveation enables efficient perceptive learning of dexterous manipulation in simulation

Nicholas M. Blauch[†] **Nicolas Moenne-Loccoz** **Jayjun Lee** **Patrick Yin** **Asif Rana**
Mohak Bhardwaj **Jessica Yin** **Sam Mady** **Marilyn Reeb** **Karl Van Wyck**
Nathan Ratliff **Jae-Hyun Jung** **Ruth Rosenholtz** **Ankur Handa** **Gavriel State**
NVIDIA
[†]nblauch@nvidia.com fovi-dex.github.io

Abstract: Advanced physics simulators and rendering engines promise to support the next-generation of robotic reinforcement learning (RL), yet training perceptive policies remains challenging due to the high-costs of rendering and perception. We propose foveation as a solution to this challenge, coupling a novel foveated rendering solution with foveated perception, alongside a careful approach for ensuring transfer to foveated sampling of real uniform cameras. We train a language-conditioned visuotactile policy controlling a dexterous arm-hand to find and grasp a target object in clutter (FOVI-Dex), via online RL without demonstrations. In comparison to a high-resolution fixed-camera control, FOVI-Dex uses 8.9× less vision tokens, supporting 6.0× more parallel environments and 4.4× higher training throughput. At matched training budgets, this efficiency enables FOVI-Dex to outperform both the high-resolution baseline and a matched uniform active camera baseline, in both sim and real, with an outsized advantage in the real world.

Keywords: Foveated rendering, Active perception, Robot learning, Sim2Real

1 Introduction

Simulation has emerged as a powerful tool for training robotic policies via reinforcement learning [1, 2, 3]. To date most approaches rely on some state-based training to reduce the costs of perception-in-the-loop training. The state-based policy can either be distilled into a perception-based policy via supervised learning [4], or combined with state-estimation using off-the-shelf perception models to substitute for privileged simulator state variables in the learned policy [5]. However, both induce a strong observation gap between perception and state variables, making sim2real transfer difficult. Depth-based training can help close this observation gap [6] for distillation into stereo RGB. However, training direct from RGB remains a critical step for leveraging the realism and scalability of simulation-based learning for general-purpose, high-performing robot policies, allowing robots to learn visuomotor representations through active experience that is well matched to what they will experience in the real world. Nevertheless, training from RGB has remained challenging, as visual perception can quickly consume GPU resources used to parallelize the simulation, slowing down learning and mandating either immense computational scale or smaller models.

Here, we offer a biologically-inspired solution for scaling perceptive RL, based on human foveated perception, which optimizes for a trade-off between resolution, field-of-view, and energy efficiency, concentrating resources in the center of gaze and progressively reducing resolution in the periphery.

2 Method

Foveation describes a biological phenomena in which the retina exhibits a non-uniform sampling of photoreceptors - densest in the center of gaze, and progressively coarser in the periphery [7] - that

Dex Manipulation Workshop, 10th Conference on Robot Learning (CoRL 2026), Austin TX, USA.

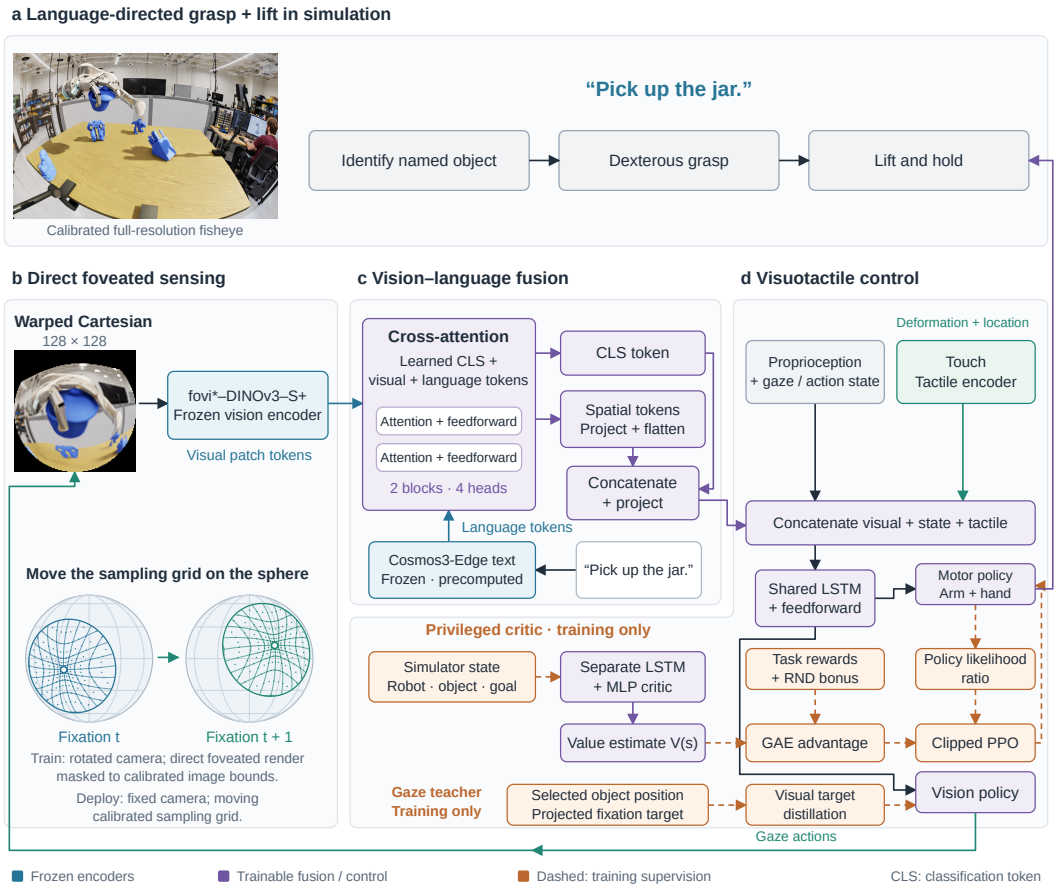


Figure 1: **Language-directed grasping with active visuotactile control.** (a) Language-directed grasp and lift in simulation. (b) Foveated rendering and fixed-camera deployment. (c–d) Cross-attention fuses frozen DINOv3 ViT-S+ features with language; tactile and proprioceptive features join them in a shared LSTM. Dashed orange paths denote training-only learning signals.

is also linked to cortical representation, where the center of gaze utilizes a disproportionate amount of cortical real estate, known as cortical magnification [8]. To model foveation in simulation, we introduce a novel form of foveated rendering to reduce the cost of high-fidelity rendering and deliver the exact set of rays needed for the model's foveated perception. Beyond reducing rendering costs, this foveation also reduces perceptual costs [9].

Direct foveated rendering. We leverage and extend the `fovi` toolbox [9], which implements multiple foveated sampling algorithms for deep learning models. First, we add support for foveated ray sampling in spherical coordinates, allowing for native transfer to foveated sampling of pixels from a calibrated fisheye cameras. Second, we opt to use the simplified `warped_cartesian` mode of foveation [10] in place of the locally isotropic FOVI model (Figure 1b), using the standard FOVI cortical-magnification function (CMF); its native 2D manifold enables us to leverage the `OmniLensDistortionLutAPI`, supporting real-time path traced rendering within RTX (see Appendix). We generate an $N \times N$ foveated array of rays by warping a Cartesian grid according to the CMF $M(r) = 1/(r + a)$, where r is angular eccentricity and $a > 0$ is the CMF parameter in the same angular units; we report a in degrees. For normalized grid coordinate $\mathbf{c}$, its angular eccentricity is

$$r(\rho) = a\left[(1 + r_{\max}/a)^{\rho} - 1\right], \qquad \rho = \|\mathbf{c}\|_2 \le 1. \tag{1}$$

Smaller a concentrates samples near fixation; $r_{\max}$ sets the circular receptive field, which we match to our calibrated camera's short side. Grid corners outside the circle are masked to prevent an

irregular warped input field resulting from the magnification function. Each sample defines a gaze-centered ray $\mathbf{d}(\mathbf{c})$. A fixation rotation R_t and calibrated fisheye projection Π give its source-image location $\mathbf{p}_t = \Pi(R_t\mathbf{d}(\mathbf{c}))$. In simulation, these rays are traced directly from the physical camera's optical center, avoiding a dense intermediate fisheye render. At deployment, software foveation samples the fixed camera image at the corresponding locations. Both paths mask samples outside the calibrated camera view, minimizing sim2real gaps.

Multimodal foveated policy architecture We pre-adapt DINOv3-S+ over a range of foveation strengths ($a \in [0.001, 0.01, 0.1, 1, 10, 100, 1000]$) using FOVI's LoRA strategy [9], with 128^2 warped inputs and 16^2 patches, and freeze it during robot learning. Semantic and spatial visual features are fused with language, proprioception, and touch in a recurrent policy (Figure 1).

Simulation environment, task, and training We train our policies in IsaacLab [3]. High-quality USDs are used to simulate the robot, table, objects, background, and lighting conditions used in our real-world experiments (Figure 1A). We introduce a challenging visual clutter task that requires a robot arm-hand (Flexiv Rizon 4s arm, Sharpa Wave W02 hand) to select and lift one of five cluttered objects based on a language prompt, via perception from a single, fixed fisheye camera. Objects are uniformly colored – without color or textural variation across the cluttered array – forcing a shape-based decision. The policy must learn to find the named object, grasp it, and lift it to the goal location as quickly as possible. We measure success-region occupancy (ISR) as the fraction of evaluation time in which the target object is within 10 cm of its elevated goal.

We use automatic domain randomization (ADR) to progressively increase robustness [2]. Levels advance when global success-region occupancy exceeds 40% for five consecutive control steps, followed by a synchronized environment reset. All models start at level zero and progress toward level 50. We train motor policies with asymmetric actor–critic PPO over dense rewards, using a privileged state critic during training, 60-hz control, 24-control-step recurrent sequences, and 4-step action chunks. Further ADR and training details are in the Appendix.

Model variants We compare three visual configurations (Figure 2A). The two active variants receive matched low-resolution images with circular fields of view and gaze supervision, differing only in foveated versus uniform angular sampling. A high-resolution uniform fixed-camera control receives the full rectangular image without a gaze policy. Both active models use the same FOVI*-DINOv3-S+ checkpoint adapted across foveation strengths; the fixed control uses the standard DINOv3-S+ checkpoint. Other model components are matched.

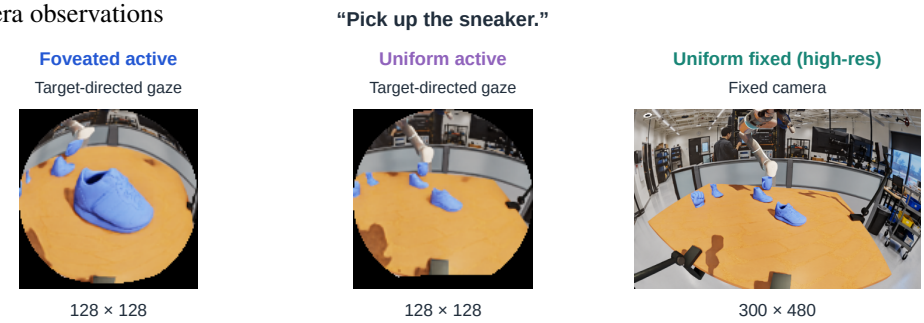
3 Results

The foveated configuration supports 6.0× more environments per GPU and 4.4× higher measured training throughput on eight GPUs than the high-resolution fixed control, with 8.9× fewer visual tokens (Figure 2B). The uniform active control has nearly the same throughput and allocation, isolating the benefit of concentrating visual detail around fixation.

Using eight GPUs, foveation improves Best ISR by 5.9 percentage points over uniform active perception. Both active models reach ADR 50 at similar times; the fixed high-resolution model takes significantly longer. With 4-GPUs, the high-resolution model does not reach ADR 50 in any seed, and only one seed reaches ISR$>$0.1. Individual-seed results are shown in Figure S2. We find that the advantage depends on the visual task demands: with one object, all models perform well, and the high-resolution control achieves the highest mean ISR (Figure S3).

To evaluate zero-shot sim2real transfer, we construct a 5-object task containing 3 similar shoe categories, procedurally generate 2000 trials with different object positions and language instructions, and use software-based active gaze from a fixed fisheye camera. We select the best-performing seed from each model type. In simulation, foveated active perception succeeds in 87.6% of trials, compared with 53.75% for uniform active and 64.2% for uniform fixed high-resolution perception; in

A Camera observations



B Allocation, training throughput, and learning

Configuration (res)	Vision Tokens (increase)	Per-GPU allocation Envs (reduction)	Mini-batch	Micro-batch	Memory (GB)	GPUs	Throughput Total FPS (slowdown)	Rollout FPS (slowdown)	Learning Best ISR at ADR 50 (%)	To ADR 50 (hours)	To Best (hours)
Foveated active	64 (1.0×)	1248 (1.0×)	7488	1248	45.2	8	20760 (1.0×)	29563 (1.0×)	61.0	**17.23**	**57.98**
						4	10762 (1.0×)	15390 (1.0×)	**61.8**	18.22	68.91
Uniform active	64 (1.0×)	1248 (1.0×)	7488	1248	45.4	8	20178 (1.0×)	28416 (1.0×)	55.1	18.01	60.59
						4	10774 (1.0×)	15420 (1.0×)	52.3	18.60	66.34
Uniform fixed (high-res)	570 (8.9×)	208 (6.0×)	1248	96	44.9	8	4685 (4.4×)	7544 (3.9×)	52.1	27.74	60.70
						4	2413 (4.5×)	3869 (4.0×)	—	—	—

C ISR vs. wall-clock time

D Simulation and real-world success

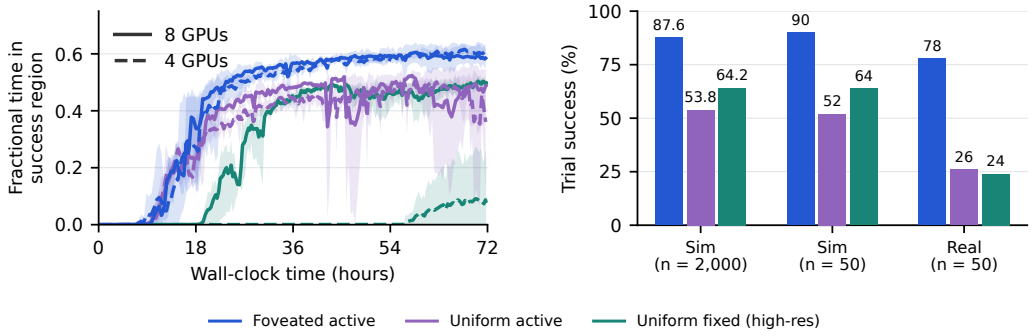


Figure 2: **Visual ablations and performance. A:** Visualization of model perceptual inputs. **B:** Memory is calibration peak (decimal GB). FPS counts control steps across all ranks, time-weighted over each complete 30,000-update L40S run, then averaged across seeds. Parentheses compare to foveated active at the same GPU count. Learning entries average three seeds: each seed's best evaluation ISR at ADR 50 within 72 hours (post-hoc early stopping), time to ADR 50, and time to that evaluation. Dashes denote incomplete attainment: no high-resolution four-GPU seed reaches ADR 50 within the analysis window. **C:** Five-object training on eight (solid) or four (dashed) L40S GPUs over a 72 hr budget. Lines are equal-weight three-seed means; bands span the seed range. **D:** Terminal trial success on the five-object, three-shoe benchmark, using the best seed per group. The first 50 simulation trials share planned layouts and targets with the 50 rated robot trials; they are a subset of the full 2,000-trial simulation evaluation.

real-world transfer, success is 78%, 26%, and 24%. Both uniform models struggle in real-world transfer; qualitatively, the uniform active model often chooses the wrong object, whereas the high-res model often fails at precise 3D localization under the challenging monocular setup. Thus, our results demonstrate a strong advantage for foveated perception in sim2real robot learning.

4 Conclusion

Foveated rendering and perception can enable direct-from-perception reinforcement learning for robotics, particularly in scenarios simultaneously requiring a wide field-of-view and fine-grained detail. The lower cost of foveated models can enable scaling the number, diversity, and complexity of simulation environments, longer horizons, and larger models, to name a few possibilities.

References

[1] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3803–3810, May 2018. doi:10.1109/ICRA.2018.8460528.

[2] OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang. Solving Rubik's Cube with a Robot Hand, Oct. 2019.

[3] NVIDIA, :, M. Mittal, P. Roth, J. Tigue, A. Richard, O. Zhang, P. Du, A. Serrano-Muñoz, X. Yao, R. Zurbrügg, N. Rudin, L. Wawrzyniak, M. Rakhsha, A. Denzler, E. Heiden, A. Borovicka, O. Ahmed, I. Akinola, A. Anwar, M. T. Carlson, J. Y. Feng, A. Garg, R. Gasoto, L. Gulich, Y. Guo, M. Gussert, A. Hansen, M. Kulkarni, C. Li, W. Liu, V. Makoviychuk, G. Malczyk, H. Mazhar, M. Moghani, A. Murali, M. Noseworthy, A. Poddubny, N. Ratliff, W. Rehberg, C. Schwarke, R. Singh, J. L. Smith, B. Tang, R. Thaker, M. Trepte, K. Van Wyk, F. Yu, A. Millane, V. Ramasamy, R. Steiner, S. Subramanian, C. Volk, CY. Chen, N. Jawale, A. V. Kuruttukulam, M. A. Lin, A. Mandlekar, K. Patzwaldt, J. Welsh, H. Zhao, F. Anes, J.-F. Lafleche, N. Moënne-Loccoz, S. Park, R. Stepinski, D. Van Gelder, C. Amevor, J. Carius, J. Chang, A. H. Chen, P. d. H. Ciechomski, G. Daviet, M. Mohajerani, J. von Muralt, V. Reutskyy, M. Sauter, S. Schirm, E. L. Shi, P. Terdiman, K. Vilella, T. Widmer, G. Yeoman, T. Chen, S. Grizan, C. Li, L. Li, C. Smith, R. Wiltz, K. Alexis, Y. Chang, D. Chu, L. J. Fan, F. Farshidian, A. Handa, S. Huang, M. Hutter, Y. Narang, S. Pouya, S. Sheng, Y. Zhu, M. Macklin, A. Moravanszky, P. Reist, Y. Guo, D. Hoeller, and G. State. Isaac Lab: A GPU-Accelerated Simulation Framework for Multi-Modal Robot Learning, 2025.

[4] T. G. W. Lum, M. Matak, V. Makoviychuk, A. Handa, A. Allshire, T. Hermans, N. D. Ratliff, and K. V. Wyk. DextrAH-G: Pixels-to-Action Dexterous Arm-Hand Grasping with Geometric Fabrics, Oct. 2024.

[5] S. Tao, J. Xu, H. Rabeti, Y. Narang, Y. Guo, and I. Akinola. SILO: Simulation-in-the-Loop Sim-to-Real Transfer for Multi-Stage Cable Routing, July 2026.

[6] R. Singh, A. Allshire, A. Handa, N. Ratliff, and K. V. Wyk. DextrAH-RGB: Visuomotor Policies to Grasp Anything with Dexterous Hands, Feb. 2025.

[7] A. B. Watson. A formula for human retinal ganglion cell receptive field density as a function of visual field location. *Journal of Vision*, 14(7):15, June 2014. ISSN 1534-7362. doi:10.1167/14.7.15.

[8] P. M. Daniel and D. Whitteridge. The representation of the visual field on the cerebral cortex in monkeys. *The Journal of Physiology*, 159(2):203–221, Dec. 1961. ISSN 0022-3751, 1469-7793. doi:10.1113/jphysiol.1961.sp006803.

[9] N. M. Blauch, G. A. Alvarez, and T. Konkle. FOVI: A biologically-inspired foveated interface for deep vision models. In *International Conference on Machine Learning*, 2026.

[10] B. Wang, D. Mayo, A. Deza, A. Barbu, and C. Conwell. On the use of Cortical Magnification and Saccades as Biological Proxies for Data Augmentation, Dec. 2021.

[11] K. Van Wyk, A. Handa, V. Makoviychuk, Y. Guo, A. Allshire, and N. D. Ratliff. Geometric fabrics: a safe guiding medium for policy learning, 2024. URL `https://arxiv.org/abs/2405.02250`.

[12] J. Lee, J. Yin, A. Rana, N. Blauch, S. Mady, M. Bhardwaj, N. Fazeli, N. Ratliff, K. Van Wyk, and A. Handa. ADEPT: Accelerating dexterity via pre-training and post-training using reinforcement learning, 2026. URL `https://arxiv.org/abs/2608.19182`.

[13] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal Policy Optimization Algorithms, Aug. 2017.

[14] T. Chen, M. Tippur, S. Wu, V. Kumar, E. Adelson, and P. Agrawal. Visual Dexterity: In-Hand Reorientation of Novel and Complex Object Shapes. *Science Robotics*, 8(84):eadc9244, Nov. 2023. ISSN 2470-9476. doi:10.1126/scirobotics.adc9244.

[15] B. Yi, C. M. Kim, J. Kerr, G. Wu, R. Feng, A. Zhang, J. Kulhanek, H. Choi, Y. Ma, M. Tancik, and A. Kanazawa. Viser: Imperative, Web-based 3D Visualization in Python, July 2025.

Appendix

A Rendering and camera implementation

A.1 Sampling grid and representational manifold.

Let r denote visual eccentricity, θ polar angle, and w the integrated cortical magnification function (CMF):

$$M(r) = \frac{1}{r+a}, \qquad w(r) = \int_0^r M(t)\,dt = \log(1 + r/a)\,. \tag{2}$$

For maximum eccentricity $r_{\max}$, define $k_a = 1/w(r_{\max})$ and normalized cortical radius $\bar{w}(r) = k_a w(r)$. We place an $N \times N$ Cartesian lattice in this normalized representational plane: $\mathbf{c}_{ij} = (\xi_i, \xi_j)$, with $\xi_i = -1 + (2i+1)/N$, $i \in \{0, \ldots, N-1\}$. At each grid point, $\bar{w} = \|\mathbf{c}_{ij}\|_2$ and $\theta = \operatorname{atan2}(c_y, c_x)$ give the visual sampling location

$$r(\bar{w}) = a\left[\exp(\bar{w}/k_a) - 1\right], \qquad (x, y) = r(\bar{w})(\cos\theta, \sin\theta). \tag{3}$$

The circular receptive field retains $\bar{w} \le 1$, equivalently $r \le r_{\max}$; square-grid corners are masked to zero. Thus N specifies the full image side, including padding. Both r and the positive CMF parameter a are in degrees; smaller a yields stronger foveation. The code's normalized visual coordinates are $\mathbf{s} = (x, y)/r_{\max}$. While this particular instantiation of foveation does introduce some warping of the sampling grid, in contrast to locally isotropic methods such as FOVI `isotropic`, its native 2D representational manifold supports higher-quality rendering via the Omniverse RTX renderer, for reasons described below, so it is adopted here. Simpler ray tracing applications can easily substitute the `isotropic` method, since a 2D manifold is not generally needed for post-processing, much of which can be handled by kNN pooling and convolution operations [9].

A.2 From image sampling to spherical rays.

For a calibrated fisheye camera, we interpret r as angular eccentricity. Each valid sample defines the unit ray

$$\mathbf{d}(\mathbf{c}) = (\sin r\cos\theta,\ -\sin r\sin\theta,\ \cos r)^\top, \tag{4}$$

in an image-camera frame with x right, y down and z forward; the central ray is $(0, 0, 1)^\top$. Angles in these equations are in radians. The retinal field $2r_{\max}$ is obtained by unprojecting the endpoints and center of a centered window along the calibrated image's short side and summing the two angular half-spans. A configurable window fraction η controls its size. For a centered ideal equidistant camera with focal length f in pixels, this reduces to $2r_{\max} = \eta \min(H, W)/f$.

The grid is fixed in fixation-centered angular coordinates; its sampling positions in the source image depend on fixation. Let Π denote the calibrated direction-to-pixel projection and $\mathbf{q}_t$ the fixation pixel. We unproject $\mathbf{q}_t$ to a unit gaze direction and construct the pan–tilt rotation R_t that aligns the retinal central ray with it:

$$\mathbf{v}_t = \Pi^{-1}(\mathbf{q}_t), \qquad R_t(0, 0, 1)^\top = \mathbf{v}_t, \qquad \mathbf{p}_{ij,t} = \Pi(R_t \mathbf{d}_{ij}). \tag{5}$$

This fixation-dependent mapping is necessary even for an ideal equidistant fisheye. Its image radius is proportional to angle about the camera's *optical axis*, whereas r describes angle about the current *gaze direction*. Away from the optical axis, rotating the retinal ray bundle generally changes both the shape and spacing of its projected grid; translating a centrally calibrated pixel grid does not reproduce these rays. The lens calibration remains fixed: only the source sampling positions and their visibility are recomputed as fixation changes.

A.3 Hybrid idealized sensing from a fixed fisheye camera.

Software foveation samples a fixed camera image at the fixation-dependent positions $\mathbf{p}_{ij,t}$ from Eq. (5). In simulation, we instead trace these directions from the same optical center: for physical

camera pose $(R_C, \mathbf{o}_C)$, a sample ray is $\mathbf{x}(\ell) = \mathbf{o}_C + \ell R_C R_t \mathbf{d}_{ij}$, $\ell \geq 0$. This renders the foveated grid directly without first producing a dense fisheye image. The physical camera pose is unchanged by gaze commands; only the virtual sampling orientation changes. A sample is retained only if it belongs to the retinal disc and its projection is within the source image, the calibrated angular domain, and any specified image circle. These checks preserve the visibility constraints of the fixed camera. Direct rendering and source-image sampling share ray geometry, although their reconstruction and interpolation operations differ. The renderer uses NVIDIA RTX, NVIDIA's ray-tracing rendering technology; it still renders rectangular padding and masks invalid samples afterward.

A.4 Omniverse RTX implementation.

We attach the `OmniLensDistortionLutAPI` schema to each rendering camera and select the `lut` lens model. Two precomputed lookup table (LUT) textures define its generalized projection. They are stored in OpenEXR (EXR), a high-dynamic-range image format supporting floating-point channels. The red, green, and blue (RGB) channels of the direction texture store one canonical ray per native grid texel, in RTX's x-right, y-up, z-backward frame. The red and green (RG) channels of the exit texture map octahedrally encoded directions back to texture coordinates by analytically inverting the angular mapping and CMF warp. This supplies both pixel-to-ray and ray-to-image correspondence. The textures remain fixed as gaze changes: the virtual rendering-camera orientation rotates the canonical ray bundle. RTX returns an image at the native render-grid dimensions; masking and a fixed transpose/flip publish the upright $N \times N$ sensor image without an irregular nearest-sample gather. Texture conventions and optional oversampling are described below.

A.5 Camera actions and idealized control.

The implemented hybrid controller maintains fixation $\mathbf{g}_t = (g_{\text{row}}, g_{\text{col}}) \in [0,1]^2$ in the fixed source image. A two-component policy action is clipped componentwise to $[-1, 1]$, then scaled and norm-limited to a maximum displacement $\delta_{\max}$. With the default persistent, unsmoothed update,

$$\mathbf{h}_t = \text{clip}(\mathbf{u}_t, -1, 1), \qquad \Delta\mathbf{g}_t = \delta_{\max} \frac{\mathbf{h}_t}{\max(1, \|\mathbf{h}_t\|_2)}, \qquad \mathbf{g}_{t+1} = \text{clip}(\mathbf{g}_t + \Delta\mathbf{g}_t, 0, 1). \quad (6)$$

We unproject $(W g_{\text{col}} - \frac{1}{2}, H g_{\text{row}} - \frac{1}{2})$ to obtain the gaze direction and its pan–tilt rotation R_t. The corresponding visual-angle delta is the difference between successive calibrated pan–tilt angles, not a constant multiple of the image-coordinate action. The idealized model applies this orientation exactly, without actuator lag or acceleration dynamics; gaze is held until the next control update. The explicit angular conversion and optional smoothing are given below.

Calibration, gaze rotation and control conventions. For eccentricity r about the camera optical axis, the ideal equidistant lens has image radius $r_{\text{img}} = fr$. Calibration also supports $D(r) = r(1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + k_4 r^8)$, followed by per-axis focal scales and principal-point offsets. Inversion is iterative, with a reprojection check. For fixation direction $\mathbf{v} = (v_x, v_y, v_z)^\top$ in the image-camera frame, the stored tilt τ and pan ψ are

$$\tau = -\operatorname{atan2}\left(v_y, \sqrt{v_x^2 + v_z^2}\right), \qquad \psi = -\operatorname{atan2}(v_x, v_z), \qquad R_t = R_y(-\psi) R_x(\tau). \quad (7)$$

Positive tilt looks upward and positive pan leftward, with no independent torsion. Mount roll supplies a fixed basis change for rays and gaze rotation. Invalid fixation masks the entire output.

The default fixation-dynamics preset sets $\delta_{\max} = 0.1$, persistence $\lambda = 1$, and no smoothing. The general supported update first computes $\mathbf{b} = \text{clip}[\mathbf{g}_0 + \lambda(\mathbf{g}_t - \mathbf{g}_0) + \Delta\mathbf{g}_t, 0, 1]$, then optionally uses $\mathbf{g}_{t+1} = \beta\mathbf{b} + (1-\beta)\mathbf{g}_t$; $\mathbf{g}_0 = (1/2, 1/2)$ is the neutral fixation. Hybrid control uses image coordinates; direct angle accumulation belongs to the separate hardware-gaze mode. Virtual pan–tilt state is recorded without moving the physical mount.

LUT construction and indexing. The direction texture has side $N_{\text{render}} = N$, with texel centers $((j + 1/2)/N_{\text{render}}, (i + 1/2)/N_{\text{render}})$. Canonical samples use native x as the first grid axis and y

as the second. The exit map therefore stores $(u, v) = ((c_y + 1)/2, (c_x + 1)/2)$. Transposition and vertical flipping yield the public image without resampling.

Inverse-texture texel centers decode octahedrally into RTX-frame directions. After undoing mount roll, eccentricity and azimuth determine $\mathbf{c} = \bar{w}(r)(\cos\theta, \sin\theta)$ using Eq. (2) and the padding inverse below. Out-of-domain directions map to $(-1, -1)$. The default exit-texture side is $N_{\text{exit}} = \max(128, 4N_{\text{render}})$, with an optional override. Both textures use 32-bit floating-point channels, authored as `rayEnterDirectionTexture` and `rayExitPositionTexture` under `omni:lensdistortion:lut:`. The analytic inverse is sampled onto a finite texture. The projection roles follow NVIDIA's generalized-camera specification.

Boundary padding and visibility. For square-grid corners outside the retinal disc, a smooth angular continuation prevents rays from wrapping around the sphere. At unbounded normalized visual radius $q = \|\mathbf{s}\|_2 > 1$,

$$r_{\text{pad}}(q) = r_{\max} + \frac{(\pi - r_{\max})(q-1)r_{\max}}{(\pi - r_{\max}) + (q-1)r_{\max}}, \tag{8}$$

which approaches π monotonically. Within the disc, $r = qr_{\max}$ is unchanged. The exit map applies the matching inverse continuation. Circular and calibrated source-camera masks are applied after rendering. Rendering cost therefore includes padding and samples removed by source visibility.

B Task, policy, and experimental protocol

B.1 Language-conditioned grasping in clutter.

All objects use a fixed blue color. Instructions identify the target by shape or category, without color cues, numbered goals, or spatial descriptions. The goal is fixed above the table, and grasp gating is disabled.

B.2 Policy details.

The active models use 128×128 images, with CMF $a = 3.483°$ for foveated sampling and $a \approx 9.67 \times 10^4$ degrees for near-uniform angular sampling. The fixed control uses a 300×480 image (height by width), sampled uniformly in the calibrated camera's image plane.

Masked cross-attention fuses visual tokens with precomputed language features. Its learned classification (CLS) token is concatenated with projected, flattened spatial features and projected to a visual embedding. A trainable tactile encoder processes tactile deformation and sensor locations. The recurrent actor receives these features together with proprioception and gaze/action state. Gaze learns through target distillation from the instructed object's projected position, rather than a PPO objective; privileged target positions are not actor inputs.

B.3 Geometric fabrics as a guiding policy medium.

We place a geometric fabric between the learned motor policy and joint control, using its reactive dynamics as a safe guiding medium for policy learning [11, 4]. Following the notation of ADEPT [12], let $\mathbf{q}_f, \dot{\mathbf{q}}_f \in \mathbb{R}^{29}$ denote the internal fabric position and velocity for the seven arm and 22 hand joints. The fabric evolves according to

$$\mathbf{M}_f(\mathbf{q}_f, \dot{\mathbf{q}}_f)\ddot{\mathbf{q}}_f + \mathbf{f}_f(\mathbf{q}_f, \dot{\mathbf{q}}_f) + \mathbf{f}_\pi(\mathbf{a}_t) = \mathbf{0}, \tag{9}$$

where $\mathbf{M}_f$ is the root-space metric, $\mathbf{f}_f$ collects the nominal geometric, repulsion, and dissipative terms, and $\mathbf{f}_\pi$ denotes policy-driven attraction, with its dependence on the fabric state implicit.

Palm-pose and hand-synergy actions. Our action space commands an absolute palm-pose target and a hand-designed grasp synergy with joint residuals. Each motor action is $\mathbf{a}_t = [\mathbf{a}_t^p; \mathbf{a}_t^r; a_t^c] \in$

$[-1, 1]^{29}$, with six palm coordinates, 22 residual coordinates, and one closure coordinate. After clipping to this interval, the decoder uses

$$\mathbf{x}_t^{p,*} = \boldsymbol{\ell}^p + \tfrac{1}{2}(\mathbf{a}_t^p + \mathbf{1}) \odot (\mathbf{u}^p - \boldsymbol{\ell}^p), \tag{10}$$

$$\mathbf{q}_t^{h,*} = \mathbf{q}_{\text{open}}^h + 0.26\,\mathbf{a}_t^r + \tfrac{1}{2}(a_t^c + 1)(\mathbf{q}_{\text{closed}}^h - \mathbf{q}_{\text{open}}^h). \tag{11}$$

Here $\mathbf{x}_t^{p,*}$ contains translation and ZYX Euler angles in the robot-base frame, bounded by the configured palm workspace $[\boldsymbol{\ell}^p, \mathbf{u}^p]$; hand residuals are in radians. The synergy continuously interpolates between the configured open and closed hand poses, while the residuals permit independent finger adjustments. The fabric converts the palm target into seven rigidly attached control-point targets and attracts the hand joints toward $\mathbf{q}_t^{h,*}$.

Reactive guidance and execution. Attractor, collision, and joint-limit terms act through their respective task maps and are pulled back into the shared joint configuration space. Body-sphere repulsion discourages specified self-collisions and collisions with the modeled table, workspace boundaries, and camera obstacles; joint-limit repulsion, postural terms, damping, and speed regulation shape the resulting motion. This guidance depends on the configured collision geometry and accurate low-level tracking; it does not certify avoidance of unmodeled contacts. At each control step, a displacement integrator advances the fabric's internal state, and its joint position and scaled velocity become the low-level controller targets. Four-step action chunks supply successive motor targets; the fabric continues to run for every control step. The same fabric interface is used in simulation and deployment, and the two gaze coordinates of active models are controlled separately.

B.4 Training allocation and objectives.

Environment counts are selected for an approximately 45 GB peak-memory budget per GPU, calibrated on one NVIDIA RTX PRO 6000 Blackwell GPU at ADR level 50. Training uses four or eight NVIDIA L40S GPUs with the same per-GPU allocations; increasing GPU count therefore also increases the global environment and batch counts. The reported memory is the calibration peak, rather than a measurement on the L40S training hardware. For each configuration, actor and critic minibatches contain $6E$ policy decisions, where E is the per-GPU environment count. Microbatches contain complete recurrent sequences, and training uses bfloat16 mixed precision.

Each condition uses seeds 43, 47, and 51 and trains for 30,000 updates. Learning comparisons use the first 72 hours; evaluation follows the current ADR level. Best ISR is selected post hoc at ADR 50, rather than measured on an independent held-out test. The one-object experiment uses eight GPUs and the same per-GPU allocations, with placements sampled from the clutter task's five-slot layout. Reported throughput excludes startup, evaluation, and checkpoint overhead.

Motor learning uses PPO [13], with hand-to-object, object-to-goal and annealed lift rewards, finger-posture regularization, and random network distillation (RND) exploration. A separate recurrent critic receives clean robot state, forces and torques, target and clutter states, the goal, hand–object distances, and controller state. Its values supply the baseline for generalized advantage estimation. An actor-side value head receives auxiliary return regression but does not supply this baseline.

C Domain randomization and curriculum

All visual variants share the schedules in Table S1 and the scene variation in Table S2. ADR advances synchronously across GPUs using the occupancy criterion described in the main text; evaluation uses the current level and does not suspend advancement. Joint dynamics are sampled uniformly and independently per environment and joint at reset and on level increases. Multipliers apply to the robot asset's nominal coefficients.

Observation noise uses episode-specific amplitudes and a persistent bias shared across joints. Active models apply the same perturbations to angular pan–tilt observations. The critic receives clean observations; actor joint-velocity and gaze-velocity inputs are disabled.

Parameter	Level 0	Level 50
Joint reset position perturbation (rad)	0	$[-0.35, 0.35]$
Joint reset velocity perturbation (rad/s)	0	$[-1, 1]$
Actor joint-position noise: maximum half-width (rad)	0	0.08
Actor joint-position bias: maximum magnitude (rad)	0	0.04
Joint stiffness multiplier	1	$[0.5, 2]$
Joint damping multiplier	1	$[0.5, 2]$
Joint friction multiplier	0	$[0, 5]$
Joint armature multiplier	0.01	$[0.01, 1]$
Arm attractor speed-cap multiplier	1	0.5
Finger-curl regularization weight	0.01	0.005
Object-to-goal reward sharpness	15	20
Lift reward weight	5	0

Table S1: **ADR schedules for the reported training runs.** The upper block gives randomization ranges and bounds on episode-specific noise amplitudes and biases. The lower block contains deterministic curriculum settings, rather than random draws. Endpoints interpolate linearly with $\alpha = \ell/50$, except that the arm attractor speed-cap multiplier is $2^{-\alpha}$.

Appearance curriculum. The available table-texture set grows approximately linearly from one texture at level zero to the full collection at level 50. The ranges in Table S2 are present from level zero. Object identities remain fixed within each environment, with unchanged scales and blue appearance.

Lighting and laboratory backgrounds. Each environment samples one of four laboratory panoramas at reset, together with local ceiling, sky, and sunlight illumination. Three ceiling emitters share a sampled color temperature; each is independently switched off with probability 0.15. Sunlight is switched off with probability 0.25, and the closed-shade panorama disables both sky and sunlight. Background brightness and tint are coupled to the sampled illumination. Lighting and backgrounds remain fixed throughout each episode.

Parameter	Distribution or range	Resampling
Camera translation, each axis	$\mathcal{U}(-0.03, 0.03)$ m	Episode reset
Camera rotation, each Euler component	$\mathcal{U}(-2, 2)$ degrees	Episode reset
Clutter layout yaw; each object's yaw	$\mathcal{U}(0, 360)$ degrees	Episode reset
Clutter layout jitter, each horizontal axis	$\mathcal{U}(-0.01, 0.01)$ m	Episode reset
Table translation, each axis	$\mathcal{U}(-0.05, 0.05)$ m	Construction only
Table yaw	$\mathcal{U}(-5, 5)$ degrees	Construction only
Table RGB tint	$[0.3, 0.6] \times [0.2, 0.4] \times [0.1, 0.2]$	Episode reset
Table roughness; specular	$[0.3, 0.9]$; $[0, 1]$	Episode reset
Table texture rotation	$\mathcal{U}(0, 360)$ degrees	Episode reset
Robot roughness; metallic; specular	$[0.2, 1]$; $[0, 0.8]$; $[0, 1]$	Episode reset
Ceiling-light brightness multiplier	$[0.08, 1]$ of nominal intensity	Episode reset, per light
Ceiling-light color temperature	$[3900, 4700]$ K	Episode reset, shared by three lights
Sky intensity	$[300, 1100]$	Episode reset
Sun azimuth; elevation	$[-50, 50]$; $[12, 65]$ degrees	Episode reset
Sun intensity	$[1000, 6000]$	Episode reset

Table S2: **Level-independent randomization.** Numeric intervals denote uniform sampling. Light intensities use renderer authoring units. Lighting draws are independent across environments, with a shared ceiling color temperature within each environment.

D Additional experimental results

D.1 Foveation strength

We vary foveation strength in the five-object task while holding the four-GPU training recipe, frozen omni-foveated encoder, image resolution, and per-GPU allocation fixed. Each of seven strengths uses seeds 43, 47, and 51. Figure S1 compares their first 72 hours with the corresponding four-GPU main-paper setting. Intermediate foveation performs best in this sweep: the mean per-seed maximum ISR at ADR 50 is 60.5% at $a = 0.604°$, versus 52.7% at the strongest foveation and 51.7–51.9% at the two near-uniform settings. The main-paper setting remains comparable at 61.8%; overlapping seed ranges do not establish a precise optimum.

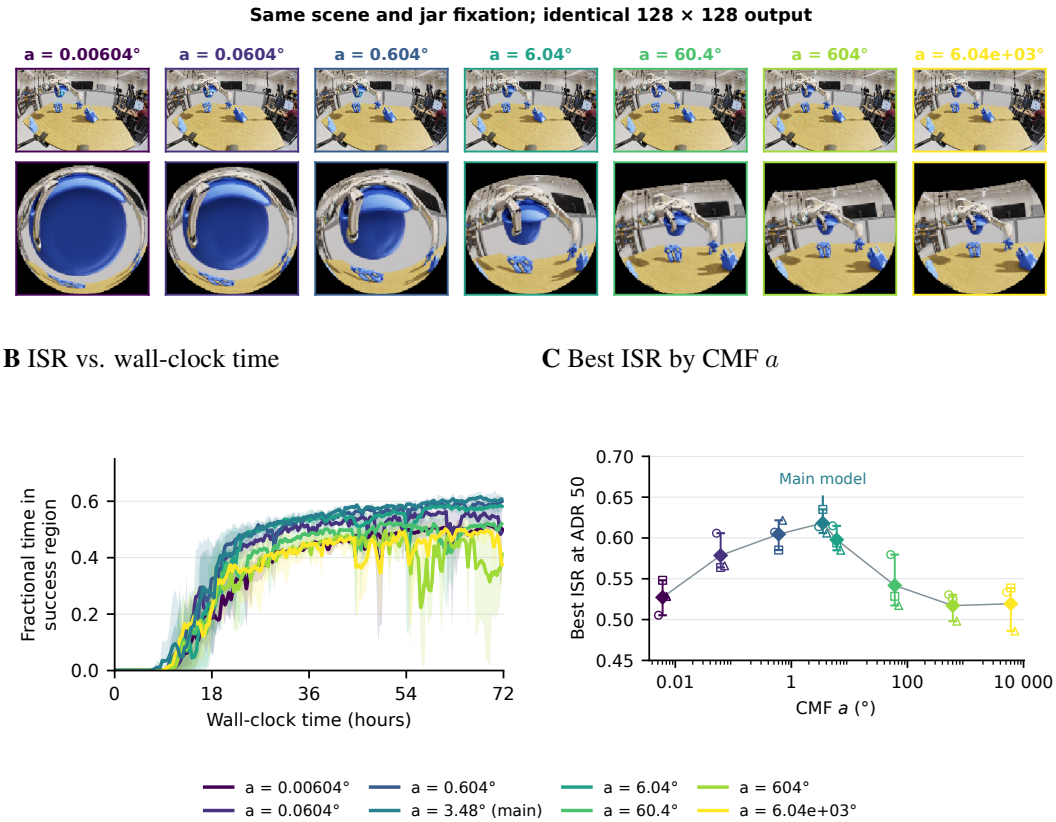


Figure S1: **Foveation-strength sweep with five objects and four GPUs. A:** A fixed simulated scene and gaze on the jar. Top: source-camera sampling footprints; bottom: corresponding 128 × 128 retinal images. These are calibrated software resamplings for illustration, not sweep-policy recordings or new simulator renders. Smaller a gives stronger foveation. Labels report CMF a in degrees, with a field-of-view diameter of 96.6973°. **B:** Equal-weight means and full min–max ranges over three seeds, matching Figure 2C. Evaluation histories are clipped at 72 hours before interpolation; means end at the last shared seed coverage. Elapsed time includes resume interruptions. **C:** Each seed's highest recorded evaluation ISR at ADR 50 within the same window, selected without smoothing or interpolation. Circles, squares, and triangles denote seeds 43, 47, and 51, respectively, with small horizontal offsets for visibility. Diamonds show their mean; bars span their range. The historical four-GPU main-paper setting, $a = 3.483°$, is included in the connected means in C and labeled "Main model." Colors follow increasing a, including the main setting, and identify the same strengths throughout.

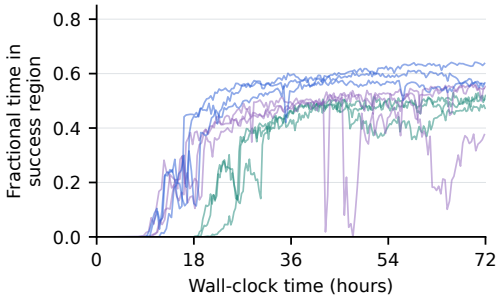
D.2 Five-object results across all seeds

Figure S2 exposes the individual seeds underlying the main comparison. All nine runs reach ADR level 50 within the shared window. Individual trajectories still vary substantially; the three-seed range in the main figure should not be interpreted as a precise estimate of population uncertainty.

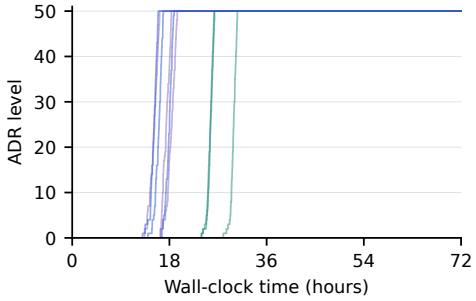
A Learning by individual seed

Configuration	Seed	Observed (hours)	Best ISR at ADR 50 (%)	To ADR 50 (hours)	To Best (hours)
Foveated active	43	72.0	57.7	18.88	44.39
Foveated active	47	72.0	61.2	16.87	59.53
Foveated active	51	72.0	64.2	15.93	70.02
Uniform active	43	72.0	53.2	16.19	42.57
Uniform active	47	72.0	56.1	18.38	69.36
Uniform active	51	72.0	56.0	19.45	69.84
Uniform fixed (high-res)	43	72.0	52.7	26.27	70.22
Uniform fixed (high-res)	47	72.0	53.0	26.36	66.02
Uniform fixed (high-res)	51	72.0	50.5	30.59	45.85

B Time in success region



C ADR level



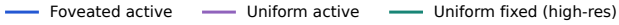


Figure S2: **Individual seeds with five objects and eight GPUs.** Each line represents one seed; seeds of the same model share color and line style, with opacity 0.5 and half the main plot's line width. Post-hoc early stopping selects each seed's highest recorded evaluation ISR at ADR 50 within the first 72 hours; "To Best" gives that evaluation's elapsed wall time. "To ADR 50" reports its first recorded attainment. Every run trains for 30,000 updates, not a fixed wall-clock duration; "Observed" reports available coverage within the analysis window, including interruptions. Curves and table statistics use the same 72-hour limit.

D.3 One-object results across all seeds

Removing distractors reduces both visual discrimination and physical clutter. All nine runs complete the curriculum, and the fixed high-resolution model achieves the highest mean post-hoc selected ISR. Thus the cluttered-task advantage of foveation does not imply a uniform ranking across task complexity. Per-GPU allocations are unchanged from the five-object experiment.

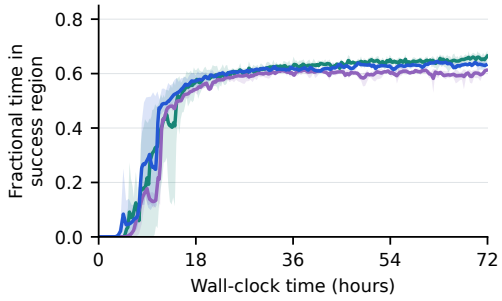
E Sim2real transfer

We deploy a foveated policy on the real robot using a fixed Basler Dart camera with a Commonlands CIL926 equidistance fisheye lens. The camera streams at 60 Hz and 1920×1200 pixels. Optimized sampling kernels extract the foveated input directly from the full-resolution image, using the calibrated projection to sample the same visual directions as the virtual camera. No physical gaze motion is required.

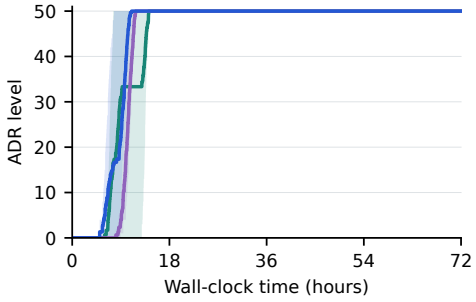
A Learning by individual seed

Configuration	Seed	Observed (hours)	Best ISR at ADR 50 (%)	To ADR 50 (hours)	To Best (hours)
Foveated active	43	72.0	65.4	10.60	69.64
Foveated active	47	72.0	65.3	11.00	62.74
Foveated active	51	72.0	64.9	7.74	64.23
Uniform active	43	72.0	61.6	11.76	70.09
Uniform active	47	72.0	63.5	11.52	55.82
Uniform active	51	72.0	62.0	10.68	71.80
Uniform fixed (high-res)	43	72.0	64.4	14.18	46.31
Uniform fixed (high-res)	47	72.0	68.4	9.26	71.97
Uniform fixed (high-res)	51	72.0	68.3	7.59	70.12

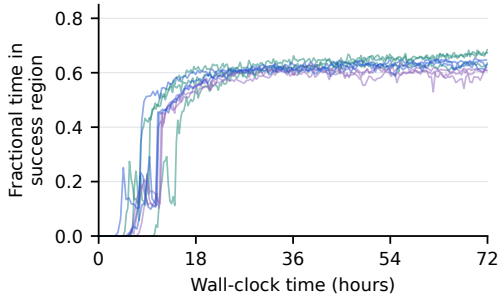
B Time in success region: mean and range



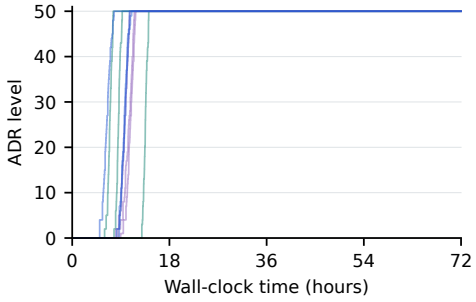
C ADR level: mean and range



D Time in success region: individual seeds



E ADR level: individual seeds



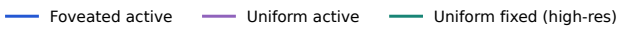


Figure S3: **One-object learning on eight GPUs.** The common plotting window is 72 hours. Table ISR uses each seed's post-hoc maximum at ADR 50 within this window. Training ends after 30,000 updates, but later observations are excluded. Other table conventions match Figure S2. B–C show equal-weight means and full ranges over three seeds; D–E show each seed separately. Evaluation follows the run's current ADR level. All models reach level 50 in every seed within this window.

Five-object, three-shoe benchmark. We evaluate the boot, classy laced boot, sneaker, fist, and cup together, with a language instruction identifying one target. The discrete layout uses a four-row, four-column grid with its corner sites removed, leaving 12 sites. The grid fits a circle centered in the usable tabletop between the camera-side Fabric collision wall and the opposite table edge, with a 5 cm clearance for object origins. The shoes have parallel 75-degree headings in the fisheye table view. For convenient physical trial ordering, one object moves to an empty site and the target changes between successive trials. All models use the same 2,000 planned simulation trials, balanced across the five targets. The first 50 share planned layouts and instructions with the robot evaluation, with ten trials per target, and form a subset of the full simulation evaluation.

We use the final checkpoints from the completed eight-GPU runs: seed 51 for foveated active and uniform active, and seed 47 for uniform fixed high-resolution perception. Each model uses identical weights in simulation and deployment, with deterministic mean actions. Simulation uses PhysX/Isaac RTX at ADR level 50, with planned object poses and targets replacing random layouts; other physics and visual randomization remain enabled. Each trial lasts at most 10 s and succeeds as soon as the instructed object enters the goal zone. Lifting a distractor causes immediate failure. Workspace and environment termination conditions remain active, with no additional reset for dropping a previously lifted target. Matching the plan does not match nuisance conditions or eliminate physical placement error.

Physical outcomes are labeled from recordings, excluding invalid or superseded attempts. All retained trials have completed ratings. Results in Figure 2D characterize these selected checkpoints and this fixed object set, rather than variation across training seeds.

Matched deployment views. Figure S4 pairs physical evaluation recordings with simulated deployments of the evaluated foveated active checkpoint. The robot uses 3D-printed VisDex objects [14]; both dashboards show the full fisheye camera, tactile feedback, and software-foveated policy input [15]. These illustrative simulated deployments use fixed full-camera rendering and software foveation, and are separate from the training-style simulation trials scored in Figure 2D.

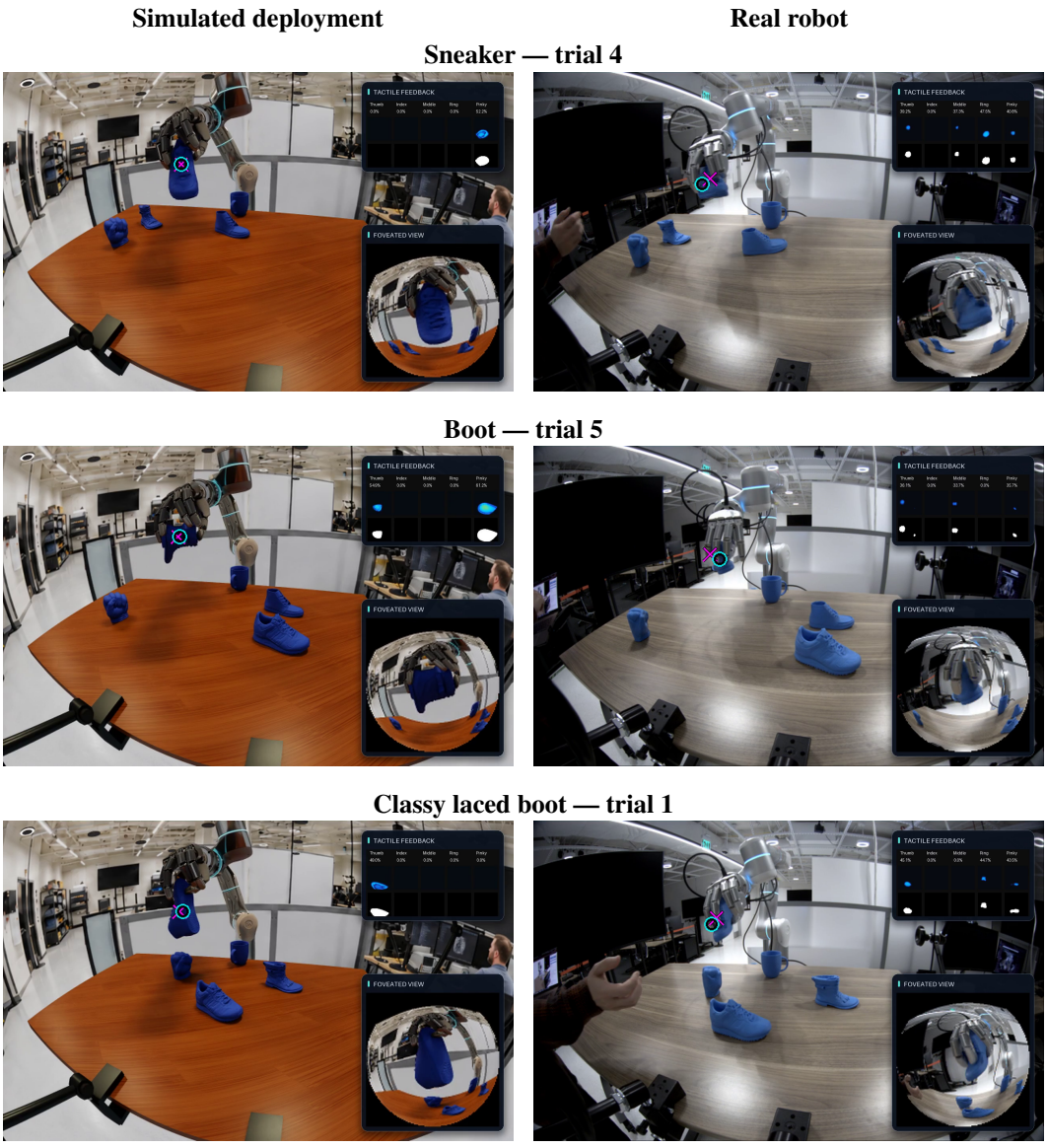


Figure S4: **Simulated deployment and robot views of the benchmark layouts.** Each row uses the same planned object arrangement, target instruction, and foveated active checkpoint in both domains. The simulated deployments are illustrative reruns, not the scored simulation trajectories. Both columns remove only the instruction header and retain the camera, tactile, and foveated views. Cyan and magenta markers indicate fixation and auxiliary target-position prediction, respectively. Frames show selected grasp/lift moments at independent video times; matching the trial plan does not imply identical nuisance conditions or manipulation trajectories.