GlimpseBC: Learning Saccadic Visual Policies from Global View Demonstrations

Anonymous authors
Paper under double-blind review

Abstract

Visual policies often process entire camera frames at every step, despite only small regions being relevant to control. In this paper, we study behavioral cloning subject to a hard pixel constraint. We introduce GlimpseBC, a simple learning algorithm in which a model jointly learns where to look and what to do from offline global view trajectories. Our model uses a differentiable crop to learn where to look from full-frame trajectories during training and a hard crop during inference, ensuring that it cannot access full-frame observations. On RoboMimic and Push-T, our models are competitive with full-frame policies (with comparable architecture) while reducing the computational cost in both training and inference. We additionally use the cluttered LIBERO-Spatial benchmark to study where narrow glimpses struggle.

1 Introduction

Many visual-policy architectures process entire camera frames uniformly at every control step. This full-frame processing can be costly in applications constrained by computation, power consumption, or perception latency Sze et al. (2017); Loquercio et al. (2021); Thavamani et al. (2021).

A substantial body of work therefore seeks to allocate visual resources selectively rather than uniformly across a scene Bajcsy (1988); Ballard (1991). One family of approaches draws inspiration from saccades—the rapid eye movements through which humans acquire task-relevant visual information. Saccadic models similarly process a sequence of small, selected glimpses rather than the entire visual field Mnih et al. (2014); Rangrej et al. (2022; 2023).

Existing approaches to selective vision in behavioral cloning (BC) often supervise visual selection with recorded human gaze Chuang et al. (2026). Furthermore, though recent methods learn visual relevance from action demonstrations, they permit some form of global context when deciding where to look Zhuang et al. (2026); Kerr et al. (2025).

We instead study BC under a hard pixel constraint: the policy receives neither fixation annotations during training nor full-frame observations at inference. This setting offers two practical benefits. First, because it requires only standard demonstrations consisting of global-view observations and expert actions, it needs no gaze-tracking hardware or additional data collection, and applies directly to existing BC datasets and pipelines. Second, because the policy never consults the full frame at deployment, its reduction in visual computation is guaranteed rather than offset by a global processing pass. The cost of these benefits is that the model must learn where to look only from its previously observed glimpses.

Related work in image classification and video recognition offer several methods to learn where to look without fixation demonstrations. One approach is to treat glimpse selection as a sequential decision problem and optimize it with reinforcement learning Mnih et al. (2014). Another is to simplify the optimization of hard attention through staged training Elsayed et al. (2019). A further strategy is to use a full-frame teacher to supervise a glimpse-only student, as in GliTr Rangrej et al. (2023). As these strategies were developed primarily for

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.

recognition and video prediction, it is unclear which of these ideas would work in the BC setting. This motivates the following question:

Under behavior cloning, how should a model learn where to look from global-view demonstrations?

Learning where to look is challenging because selecting a hard image crop is a discrete decision and therefore not directly differentiable. Surprisingly, we find that in the behavioral-cloning setting, a simple differentiable relaxation is sufficient. In GlimpseBC, we replace the hard crop with a differentiable sampler during training, allowing the standard behavioral-cloning loss to jointly learn both where to look and what action to take. At inference, we restore the hard crop, so the deployed policy observes only the selected image region. Despite its simplicity, GlimpseBC also outperforms an adaptation of GliTr, a state-of-the-art saccadic model for online video action prediction, on several behavioral-cloning tasks.

Though GlimpseBC is often effective, it can struggle when the visual budget becomes very small. The key difficulty is that, if a narrow crop initially misses the action-relevant region, the action loss indicates that the prediction is incorrect but provides little information about where the crop should move. The second ingredient of GlimpseBC is therefore field-of-view annealing, which begins training with a broader crop and gradually shrinks it to the target deployment size. This improves the optimization of narrow glimpses while leaving the inference-time visual budget unchanged.

Because robotic systems often operate under constrained visual and computational budgets, we evaluate GlimpseBC on three robotic manipulation benchmarks. We use four tasks from RoboMimic, the Push-T task, and ten tasks from LIBERO-Spatial and compare against full-frame policies, fixed-crop policies, and GliTr. GlimpseBC consistently outperforms GliTr on RoboMimic and Push-T, while the results on LIBERO-Spatial are mixed. At moderate crop sizes, it also matches or exceeds comparable full-frame policies on several tasks while reducing computation during both training and inference. We ablate the downstream policy architecture, sampling geometry, and visual budget.

2 Related Work

Selective Vision in Imitation Learning. Several imitation-learning methods use recorded human gaze to supervise where a policy should attend Zhang et al. (2018); Saran et al. (2021); Kim et al. (2020; 2021; 2022); Chuang et al. (2026). Pal et al. compare an attention model trained with human gaze against one learned jointly with a driving policy Mohandoss et al. (2018). Seeker learns regions of interest from action supervision, but predicts them from frozen features of the current full image Zhuang et al. (2026). EyeRobot learns a gaze policy through reinforcement learning in a rendering environment constructed from demonstrations recorded with a 360-degree camera Kerr et al. (2025).

GlimpseBC instead needs none of gaze labels, current-frame global features, or a reconstructed environment.

Hard Attention for Image Tasks. Hard-attention models process a sequence of selected image regions. RAM introduced the canonical recurrent architecture, in which a policy selects multi-resolution glimpses, updates a hidden state, and predicts its next fixation Mnih et al. (2014). Ba et al. extended this approach to weakly supervised multiple-object recognition Ba et al. (2015). Later work improves the training and scalability of hard-attention models Elsayed et al. (2019); Papadopoulos et al. (2021); Rangrej & Clark (2021); Wang et al. (2020). Other approaches enforce consistency across sequential glimpses Rangrej et al. (2022) or use reinforcement learning to select glimpses with variable position and scale Pardyl et al. (2024). These papers focus mainly on recognition from image-level labels rather than behavioral cloning from expert actions.

Adaptive Visual Computation for Images and Video. Spatial transformer networks introduced a differentiable module for learning spatial transformations of an input feature map Jaderberg et al. (2015). AdaFocus V2 uses a related differentiable interpolation operation

to train patch selection for video recognition Wang et al. (2022a). FOVEA resamples high-resolution images into smaller canvases for object detection Thavamani et al. (2021), while SaccadeCam redistributes spatial resolution under a fixed camera bandwidth Tilmon & Koppal (2021). Policy-based foveated imaging similarly learns task-aware pixel acquisition Xiao et al. (2026). These approaches either access a broad current observation before selecting a region or distribute sensing capacity across the current field of view. GlimpseBC instead selects its next crop only from the glimpses observed so far. While GlimpseBC uses the same differentiable-sampling mechanism, our contribution is showing that in behavioral cloning, the per-timestep action loss alone is sufficient to train it under a strict glimpse-only inference constraint, whereas feature distillation from a full-frame teacher is not.

Related approaches allocate computation across the spatial and temporal dimensions of video. Some use soft attention across frames Sharma et al. (2015) or restrict expensive processing to selected regions. AdaFocus and its extensions learn where and when to apply computation Wang et al. (2021; 2022b; 2025). Related methods use recurrent, reinforcement-learned, or glance-and-focus attention Chen et al. (2020); Huang et al. (2023); Wang et al. (2017), as well as adaptive spatio-temporal sampling or pose-guided selection Mac et al. (2023); Baradel et al. (2017); Das et al. (2019). These methods address efficient recognition rather than learning a control policy from expert demonstrations.

Constrained Video Models. Some video models explicitly limit what the model observes. Yeung et al. train a recurrent action detector to select which frames to process Yeung et al. (2016); this restricts temporal access, but each selected observation remains a complete frame. Glimpse Clouds instead tracks sparse, activity-relevant points over time using recurrent workers and external memory Baradel et al. (2018).

GliTr Rangrej et al. (2023), the closest method, also selects glimpses only from past glimpses, but trains its fixation policy by matching a full-frame teacher's features and predictions; GlimpseBC instead trains it jointly with the action policy from the behavioral-cloning objective.

3 GlimpseBC

3.1 Architecture

GlimpseBC consists of two learned components: a fixation predictor that determines where to look and an action policy that predicts the expert action from the resulting visual representation. Because GlimpseBC only modifies the visual encoder, it can be paired with different downstream action-policy architectures.

The underlying idea of the GlimpseBC encoder is to replace the standard full-frame visual encoder with a saccadic encoder that retains information from previously observed glimpses. At timestep t, the model selects a normalized fixation location $\ell_t \in [0,1]^2$, which specifies the crop's top-left corner, and extracts a $P \times P$ crop from the current RGB frame I_t. We denote this crop by $x_t = C(I_t, \ell_t)$.

We encode x_t with a ResNet-18 and linearly project the resulting feature to the encoder dimension d. We separately project the fixation location ℓ_t into the same dimension and add the two representations to form the glimpse token

$$z_t = W_{\text{img}} f_{\text{ResNet}}(x_t) + W_{\text{loc}} \ell_t \in \mathbb{R}^d.$$

To retain information across timesteps within the current observation window, we process the accumulated glimpse tokens with a temporal Transformer:

$$M_{1:t} = \mathcal{T}(z_1, \ldots, z_t) = (m_1, \ldots, m_t)$$

so that at timestep t, the Transformer operates only on the observed prefix $z_{1:t}$. The encoder uses this visual memory both to determine the next glimpse and to form the representation used for action prediction. For action prediction, we concatenate the current Transformer representation from each camera with an encoding of the corresponding low-dimensional

robot state s_t. The output MLP then produces the observation feature supplied to the downstream action policy

$$e_t = g_{\text{out}}\left(\left[\underset{\text{cameras}}{\text{concat}}(m_t)\,;\, f_{\text{state}}(s_t)\right]\right), \qquad \hat{\mathbf{a}} = \pi(e_{1:t}).$$

For each camera, the model begins from a learned initial fixation and predicts subsequent fixations from the latest memory token given by $\ell_{t+1} = g_{\text{loc}}(m_t)$ where the location head g_{loc} consists of layer normalization and a two-layer MLP with a GELU activation and final sigmoid. The complete architecture is depicted in Figure 1.

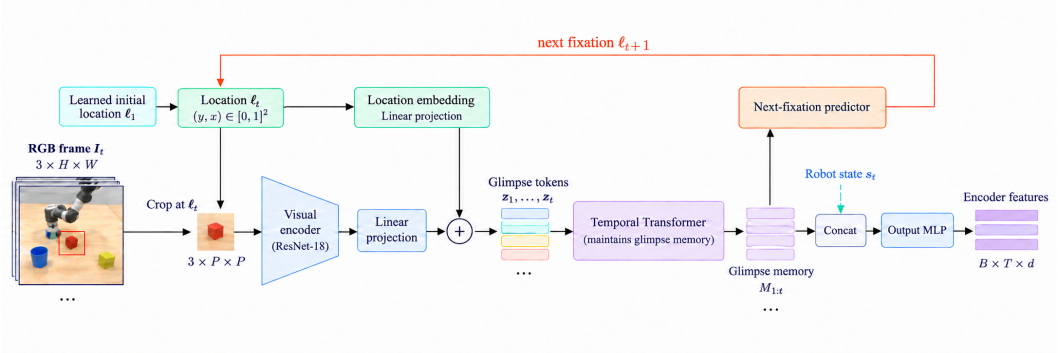


Figure 1: GlimpseBC encoder. At timestep t, the action a_t and next fixation ℓ_{t+1} depend only on the observed prefix $(z_1, \ldots, z_t)$.

3.2 Training

We train the GlimpseBC encoder and downstream action policy jointly using the policy's usual behavioral-cloning objective:

$$\mathcal{L}_{\text{GlimpseBC}}(\theta, \phi) = \mathcal{L}_{\text{BC}}\left(\pi_\phi(E_\theta(o)), \mathbf{a}^\star\right).$$

Here, E_θ is the GlimpseBC encoder and π_ϕ is the downstream action policy. We use regression for the RNN policy and flow matching Lipman et al. (2023) with the ChiUNet architecture from Diffusion Policy Chi et al. (2025). There is no separate loss on the fixation locations; the action-loss gradient reaches the fixation predictor through the differentiable crop. During training, we maintain exponential-moving-average copies of the encoder and policy weights with decay 0.995 for evaluation.

Differentiable glimpse extraction. To learn where to look using the behavioral-cloning objective, the crop operation must be differentiable with respect to the predicted fixation. Let $I_t \in \mathbb{R}^{3\times H\times W}$ be the current image, $\ell_t = (\ell_t^y, \ell_t^x) \in [0, 1]^2$ the normalized top-left crop location, S the width of the source region, and P the output patch width. The corresponding top-left image coordinates are

$$u_t = \ell_t^y(H - S), \qquad v_t = \ell_t^x(W - S).$$

We place a regular $P \times P$ sampling grid over the resulting $S \times S$ source region. Each output pixel is obtained by bilinearly interpolating the neighboring pixels of the original image:

$$x_t[c, i, j] = \sum_{p,q} I_t[c, p, q]\, \kappa(p - y_{ij})\, \kappa(q - x_{ij}), \qquad \kappa(r) = \max(0, 1 - |r|),$$

where (y_{ij}, x_{ij}) denotes the continuous image coordinate associated with output position (i, j). Because these sampling coordinates vary continuously with ℓ_t, the action loss provides a gradient to the fixation predictor.

Field of View Annealing. Small crops can provide weak early gradients when an inaccurate initial fixation excludes the evidence needed to predict the action. We therefore begin with a wider source region of width S_0 and contract it to the target width $S_T = P$. At training step k,

$$S_k = S_0 \left(\frac{P}{S_0} \right)^{\min(k/K,1)},$$

where K is the annealing duration. Each $S_k \times S_k$ region is resampled to $P \times P$, keeping the encoder input fixed while visual context narrows. The schedule therefore provides broad context early and progressively requires the model to localize task-relevant regions. We let k count optimizer updates and set $K = 100{,}000$ within 200,000 total updates, after which $S_k = P$.

4 Experiments

We evaluate GlimpseBC on three datasets: RoboMimic, Push-T, and LIBERO-Spatial.

1. **RoboMimic.** RoboMimic Mandlekar et al. (2021) provides manipulation demonstrations for imitation learning and offline reinforcement learning. We use Lift, Can, Square, and Tool Hang. Performance is episodic success rate over 100 evaluation episodes with evaluation seed 10,000.
2. **Push-T.** Push-T Florence et al. (2021) requires a circular end-effector to push a T-shaped object into a target pose. We report normalized peak overlap: the maximum over an episode of the T's coverage of the target region, divided by the 95% success threshold and clipped to 1 Chi et al. (2025), averaged over 100 evaluation episodes with evaluation seed 10,000.
3. **LIBERO-Spatial.** LIBERO-Spatial Liu et al. (2023) contains ten manipulation tasks that vary the spatial arrangement of the objects. Each task requires the robot to place a black bowl on a plate in a cluttered scene. Performance is episodic success rate over 50 evaluation episodes per task.

RoboMimic and Push-T are our primary benchmarks, on which we run the full set of comparisons. We use LIBERO-Spatial as a secondary, more cluttered setting to probe where narrow glimpses fail and to study field-of-view annealing. Unless stated otherwise, all models, including the full-frame baselines, share the same training protocol of 200,000 updates and batch size 32.

4.1 Does GlimpseBC learn where to look?

To evaluate whether GlimpseBC learns where to look, we compare it against a policy where the visual input is a fixed, centered crop, but is otherwise identical to GlimpseBC (it has the same crop size, ChiUNet policy, training budget, evaluation protocol, and nine flow-matching sampling steps). This comparison is presented in Table 1.

Method	Crop	Can	Lift	Square	Tool Hang	Push-T
Fixed center	32 px	67	100	31	40	52.29
GlimpseBC	32 px	99	100	88	62	70.63
Fixed center	48 px	82	100	73	36	62.12
GlimpseBC	48 px	96	100	91	38	79.31
Fixed center	64 px	96	100	88	32	58.68
GlimpseBC	64 px	98	100	91	44	65.88

Table 1: Fixed center-crop versus GlimpseBC with nine-step ChiUNet. RoboMimic reports success rate (%) and Push-T reports normalized overlap (%). Complete 9/3/1 sampling-step flow matching results appear in Appendix A.3.2.

Across the 15 tasks and crop-size settings, GlimpseBC has a higher point estimate in 12 of them and matches the fixed crop in 3. The largest gains occur at 32 pixels, where Can improves from 67 to 99, Square from 31 to 88, and Tool Hang from 40 to 62; the three ties are on Lift, where every crop succeeds. As crop size decreases, the gap between GlimpseBC and fixed center increases. This matches the intuition that a static central crop misses off-center targets as the field of view shrinks, whereas active attention dynamically tracks them.

4.2 How does GlimpseBC perform against full-frame and prior work baselines?

Table 2 compares the complete GlimpseBC and GliTr systems at the same crop sizes. We use the official GliTr implementation and make only the minimal changes needed to apply it to behavioral cloning. Implementation details appear in Appendix D.

Method	Crop	Can	Lift	Square	Push-T	LIBERO
Full-frame RNN	Full	80	99	70	79.01	62.8
Full-frame ChiUNet	Full	98	100	91	59.0	85.2
GliTr teacher	Full	93	99	79	77.1	78.2
GliTr	32 px	0	2	11	11.7	28.2
GlimpseBC	32 px	99	100	88	70.63	32.0
GliTr	48 px	72	3	41	24.7	42.0
GlimpseBC	48 px	96	100	91	79.31	39.2
GliTr	64 px	85	6	68	48.6	54.2
GlimpseBC	64 px	98	100	91	65.88	56.7

Table 2: RoboMimic and LIBERO-Spatial report success (%). Push-T reports normalized overlap (%). LIBERO 32- and 64-pixel averages use the same nine completed tasks; all others use ten. Task-level results are in Tables 11, 14, and 22. ChiUNet entries use nine sampling steps; LIBERO GlimpseBC entries use the RNN policy. The two methods utilize their own policies and training pipelines. GlimpseBC Tool Hang results appear in Table 1.

On RoboMimic and Push-T, GlimpseBC outperforms GliTr at every shared crop size, with especially large gains on Lift, Square, and Push-T. Lift is particularly diagnostic: the full-frame GliTr teacher reaches 99% success, and even a fixed 32-pixel center crop reaches 100%, yet the GliTr student reaches only 2–6%. We hypothesize that this reflects a mismatch between GliTr's training mechanism and the supervision available in behavioral cloning. GliTr learns its glimpse policy indirectly through a full-frame teacher and auxiliary feature- and prediction-matching objectives. In BC, however, every observation is paired with an expert action, providing a direct, task-aligned signal for whether the selected glimpse contains the information needed for control. GlimpseBC optimizes fixation directly through this action-prediction signal, whereas the adapted GliTr retains auxiliary objectives designed for the video setting.

Against the full-frame ChiUNet baseline, GlimpseBC matches performance on Can, Lift, and Square and attains a higher point estimate on Push-T at 48 pixels (79.31 versus 59.0). This advantage on Push-T does not hold uniformly across downstream policy architectures. On LIBERO-Spatial, GlimpseBC is broadly comparable to GliTr, performs substantially worse than the full-frame ChiUNet policy, and remains below the full-frame RNN baseline. We postulate that its clutter and varied object layouts make it easier for a narrow initial glimpse to miss the action-relevant region, which motivates the field-of-view annealing experiments in Section 5.

4.3 The Compute–Performance Tradeoff

Table 3 reports the controlled ChiUNet compute–performance comparison with nine and one flow-matching sampling steps.

(a) Nine-step compute.

		Performance (%)		Training TFLOPs/run		Inference GFLOPs/input	
Dataset	Crop	Fixed	GlimpseBC	Fixed	GlimpseBC	Fixed	GlimpseBC
RoboMimic	32 px	59.50	87.25	8,697.0	9,228.6	1.687	1.707
RoboMimic	48 px	72.75	81.25	17,905.1	18,591.2	2.174	2.195
RoboMimic	64 px	79.00	83.25	25,384.6	26,287.1	2.575	2.596
Push-T	32 px	52.29	70.63	5,857.8	6,128.7	1.536	1.547
Push-T	48 px	62.12	79.31	10,458.8	10,806.9	1.780	1.790
Push-T	64 px	58.68	65.88	14,196.1	14,652.3	1.980	1.991

(b) One-step Push-T performance and inference compute.

Crop	Fixed overlap (%)	GlimpseBC overlap (%)	Fixed infer	GlimpseBC infer
32 px	47.09	69.00	0.303	0.313
48 px	64.87	71.41	0.547	0.557
64 px	60.96	68.97	0.747	0.758

Table 3: Fixed center-crop versus GlimpseBC ChiUNet performance and compute with nine steps (a) and one step (b). Training uses TFLOPs/run and inference uses GFLOPs/input. RoboMimic is the four-task macro mean and Push-T is normalized overlap.

At a fixed patch width, the fixation predictor adds at most 1.2% inference and 6.1% training compute relative to the fixed-crop encoder, and with one sampling step GlimpseBC remains ahead at all three Push-T crop widths. Against the full frame, the savings are substantial: annealed 48-pixel GlimpseBC nearly matches the full-frame Push-T RNN (78.53 versus 79.01) with 58% of its inference compute (Table 19). Because the full frame is never processed at deployment, these savings are guaranteed.

4.4 Architectural and Geometric Ablations

On Push-T, we test whether learned fixation depends on the downstream policy (RNN regression or ChiUNet flow matching), sampling geometry, or crop width. Complete results and compute appear in Appendix A. For geometry, the square variant is our standard square crop. The foveated variant is a circular view with a sharp center (40% of the outer radius) and a Gaussian-blurred periphery ($\sigma = 2$ output pixels).

Setting	Variant	32 px	48 px	64 px
RNN regression	Fixed center	47.55	54.09	57.83
RNN regression	GlimpseBC	50.27	76.56	73.22
ChiUNet flow matching	Fixed center	52.29	62.12	58.68
ChiUNet flow matching	GlimpseBC	70.63	79.31	65.88
Sampling geometry	Square	50.27	76.56	73.22
Sampling geometry	Foveated	56.18	72.81	73.91

Table 4: Push-T normalized peak overlap (%) across policy and sampling-geometry ablations.

GlimpseBC outperforms the matched fixed-center policy at all three crop widths with both the RNN and ChiUNet policies, so the gain is not specific to one downstream architecture. The geometry result is mixed: foveation improves the 32- and 64-pixel point estimates (and the 24-pixel estimate Table 8), while the square crop is better at 48 pixels.

5 Field-of-View Annealing

Field-of-view annealing targets the LIBERO-Spatial failure mode in Section 4, where a narrow initial crop can miss the relevant objects. We evaluate the geometric schedule from

Section 3 on Push-T and LIBERO-Spatial with the RNN policy. All schedules shrink from S_0 to the deployment width S_T for 100,000 updates. Rollout comparisons hold S_T fixed.

5.1 Schedule Selection

For each task and deployment width, we select the candidate S_0 with the lowest held-out normalized action mean-squared error (N-MSE). We use an 80/20 split of demonstration rollouts for training and validation. Only the selected schedule is then evaluated with environment rollouts.

Searching over initial widths requires training a separate policy for every task, target width, and candidate start width, so we restrict the LIBERO-Spatial search to tasks 1–3. Table 5 reports the N-MSE values used for selection and the remaining validation metrics appear in Appendix B.

(a) Push-T.

S_T	$S_0 = 48$	$S_0 = 72$	$S_0 = 96$	Selected
24	0.025726	**0.023068**	0.024223	72
32	0.023999	0.022810	**0.021688**	96
48	–	0.022612	**0.021090**	96
64	–	0.023463	**0.021434**	96

(b) LIBERO-Spatial tasks 1–3.

Task	Target S_T	$S_0 = 64$	$S_0 = 96$	$S_0 = 128$	Selected S_0
1	32	0.077804	**0.073074**	0.079789	96
1	48	**0.073814**	0.079631	0.077997	64
1	64	–	0.077329	**0.071642**	128
1	96	–	–	**0.083373**	128 (only)
2	32	0.070098	**0.066109**	0.066229	96
2	48	0.071335	**0.070074**	0.071409	96
2	64	–	**0.073061**	0.074320	96
2	96	–	–	**0.068073**	128 (only)
3	32	**0.066799**	0.073521	0.078366	64
3	48	0.076817	0.076148	**0.075652**	128
3	64	–	0.068394	**0.067723**	128
3	96	–	–	**0.069993**	128 (only)

Table 5: Held-out action N-MSE used to select S_0 for each task and target S_T. Bold marks the selected candidate. Dashes mark start widths $S_0 \leq S_T$, which are not valid annealing schedules.

5.2 Rollout Evaluation

Table 6 compares each selected or sole evaluated schedule with a model trained directly at the same target crop width.

On Push-T, the selected schedule improves over fixed-crop training at all four target widths, with the largest gains at 24 and 32 pixels. Appendix C reports the corresponding compute. On LIBERO-Spatial, annealing raises mean success across the twelve task–width settings from 55.8% to 62.0%, improving seven, matching one, and regressing four. The largest gains occur where fixed-width training nearly fails (task 1 at 32 pixels, 0 to 24; task 2 at 48 pixels, 12 to 48), consistent with annealing preventing a narrow initial glimpse from missing the objects, whereas all four regressions occur where fixed-width training already reaches at least 50%.

This is consistent with the intended role of field-of-view annealing: it is most useful when a narrow initial view makes fixation learning difficult. More generally, the field-of-view schedule is a training hyperparameter of GlimpseBC, with fixed-width training corresponding to the trivial schedule in which the field of view begins at the deployment width. Nontrivial annealing is therefore not expected to improve every setting, but rather to provide a mechanism for recovering useful fixation gradients when the target view is too narrow to optimize directly.

(a) Push-T.

$S_0 \to S_T$ (px)	No annealing	Annealed	Δ
$72 \to 24$	45.72	57.59	+11.87
$96 \to 32$	50.27	62.59	+12.32
$96 \to 48$	76.56	78.53	+1.97
$96 \to 64$	73.22	74.56	+1.34

(b) LIBERO-Spatial tasks 1–3.

Task	$S_0 \to S_T$ (px)	No annealing	Annealed	Δ
1	$96 \to 32$	0	24	+24
1	$64 \to 48$	66	66	0
1	$128 \to 64$	56	74	+18
1	$128 \to 96$	74	72	−2
2	$96 \to 32$	50	0	−50
2	$96 \to 48$	12	48	+36
2	$96 \to 64$	72	96	+24
2	$128 \to 96$	82	96	+14
3	$64 \to 32$	40	68	+28
3	$128 \to 48$	54	72	+18
3	$128 \to 64$	84	54	−30
3	$128 \to 96$	80	74	−6

Table 6: Rollouts for validation-selected or sole schedules. Push-T reports normalized peak overlap (%), LIBERO-Spatial reports success (%), and Δ is in percentage points.

6 Conclusion

We introduced GlimpseBC, which learns where to look and what to do from global-view demonstrations, choosing each crop at inference only from previous glimpses. It matches or outperforms the fixed-center policy in all 15 comparisons and outperforms GliTr at every shared crop size on RoboMimic and Push-T, showing that expert actions alone can train an effective glimpse policy. Field-of-view annealing further improves narrow glimpses without changing the deployment-time crop size. Evaluating such algorithms on physical robots remains future work.

Reproducibility Statement

The GlimpseBC architecture, differentiable crop, and field-of-view annealing schedule are specified in Section 3. Section 4 gives the training budget, batch size, seeds, EMA decay, and evaluation protocol for each benchmark. Appendix D describes our adaptation of the official GliTr implementation, and Appendix B reports every validation metric used to select annealing schedules. Appendices A and C report task-level results and training and inference compute for all main-text tables. All experiments use the public RoboMimic, Push-T, and LIBERO-Spatial benchmarks. Code will be released upon publication.

Ethics Statement

This work uses only publicly available simulated manipulation benchmarks and involves no human subjects, personal data, or newly collected demonstrations. Because GlimpseBC learns where to look from standard demonstrations, it removes the need to record human gaze. Reducing visual computation may help deploy robot policies on low-power hardware. As with any imitation-learning method, policies inherit the behavior in their demonstrations and should be validated before deployment on physical robots.

AI Use Statement

We used generative AI tools to help polish the manuscript and to help implement experiments and test ideas. The authors reviewed all AI-assisted code, results, and text, and take full responsibility for the paper's content and conclusions.

References

Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. Multiple object recognition with visual attention. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.7755.

Ruzena Bajcsy. Active perception. *Proc. IEEE*, 76(8):966–1005, 1988. doi: 10.1109/5.5968. URL https://doi.org/10.1109/5.5968.

Dana H. Ballard. Animate vision. *Artif. Intell.*, 48(1):57–86, 1991. doi: 10.1016/0004-3702(91)90080-4. URL https://doi.org/10.1016/0004-3702(91)90080-4.

Fabien Baradel, Christian Wolf, and Julien Mille. Human action recognition: Pose-based attention draws focus to hands. In *2017 IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2017, Venice, Italy, October 22-29, 2017*, pp. 604–613. IEEE Computer Society, 2017. doi: 10.1109/ICCVW.2017.77. URL https://doi.org/10.1109/ICCVW.2017.77.

Fabien Baradel, Christian Wolf, Julien Mille, and Graham W. Taylor. Glimpse clouds: Human activity recognition from unstructured feature points. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 469–478. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00056. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Baradel_Glimpse_Clouds_Human_CVPR_2018_paper.html.

Guangyi Chen, Jiwen Lu, Ming Yang, and Jie Zhou. Learning recurrent 3d attention for video-based person re-identification. *IEEE Trans. Image Process.*, 29:6963–6976, 2020. doi: 10.1109/TIP.2020.2995272. URL https://doi.org/10.1109/TIP.2020.2995272.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *Int. J. Robotics Res.*, 44(10-11):1684–1704, 2025. doi: 10.1177/02783649241273668. URL https://doi.org/10.1177/02783649241273668.

Ian Chuang, Jinyu Zou, Andrew Lee, Dechen Gao, and Iman Soltani. Look, focus, act: Efficient and robust robot learning via human gaze and foveated vision transformers, 2026. URL https://arxiv.org/abs/2507.15833.

Srijan Das, Arpit Chaudhary, François Brémond, and Monique Thonnat. Where to focus on for human action recognition? In *IEEE Winter Conference on Applications of Computer Vision, WACV 2019, Waikoloa Village, HI, USA, January 7-11, 2019*, pp. 71–80. IEEE, 2019. doi: 10.1109/WACV.2019.00015. URL https://doi.org/10.1109/WACV.2019.00015.

Gamaleldin F. Elsayed, Simon Kornblith, and Quoc V. Le. Saccader: Improving accuracy of hard attention models for vision. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 700–712, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/8dd48d6a2e2cad213179a3992c0be53c-Abstract.html.

Pete Florence, Corey Lynch, Andy Zeng, Oscar A. Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In Aleksandra Faust, David Hsu, and Gerhard Neumann (eds.), *Conference on Robot Learning, 8-11 November 2021, London, UK*, volume 164 of *Proceedings of Machine Learning Research*, pp. 158–168. PMLR, 2021. URL https://proceedings.mlr.press/v164/florence22a.html.

Gao Huang, Yulin Wang, Kangchen Lv, Haojun Jiang, Wenhui Huang, Pengfei Qi, and Shiji Song. Glance and focus networks for dynamic visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(4):4605–4621, 2023. doi: 10.1109/TPAMI.2022.3196959. URL https://doi.org/10.1109/TPAMI.2022.3196959.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 2017–2025, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/33ceb07bf4eeb3da587e268d663aba1a-Abstract.html.

Justin Kerr, Kush Hari, Ethan Weber, Chung Min Kim, Brent Yi, Tyler Bonnen, Ken Goldberg, and Angjoo Kanazawa. Eye, robot: Learning to look to act with a bc-rl perception-action loop, 2025. URL https://arxiv.org/abs/2506.10968.

Heecheol Kim, Yoshiyuki Ohmura, and Yasuo Kuniyoshi. Using human gaze to improve robustness against irrelevant objects in robot manipulation tasks. *IEEE Robotics Autom. Lett.*, 5(3):4415–4422, 2020. doi: 10.1109/LRA.2020.2998410. URL https://doi.org/10.1109/LRA.2020.2998410.

Heecheol Kim, Yoshiyuki Ohmura, and Yasuo Kuniyoshi. Gaze-based dual resolution deep imitation learning for high-precision dexterous robot manipulation. *IEEE Robotics Autom. Lett.*, 6(2):1630–1637, 2021. doi: 10.1109/LRA.2021.3059619. URL https://doi.org/10.1109/LRA.2021.3059619.

Heecheol Kim, Yoshiyuki Ohmura, and Yasuo Kuniyoshi. Memory-based gaze prediction in deep imitation learning for robot manipulation. In *2022 International Conference on Robotics and Automation, ICRA 2022, Philadelphia, PA, USA, May 23-27, 2022*, pp. 2427–2433. IEEE, 2022. doi: 10.1109/ICRA46639.2022.9812087. URL https://doi.org/10.1109/ICRA46639.2022.9812087.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: benchmarking knowledge transfer for lifelong robot learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Antonio Loquercio, Elia Kaufmann, René Ranftl, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. Learning high-speed flight in the wild. *Sci. Robotics*, 6(59), 2021. doi: 10.1126/SCIROBOTICS.ABG5810. URL https://doi.org/10.1126/scirobotics.abg5810.

Khoi-Nguyen C. Mac, Minh N. Do, and Minh P. Vo. Efficient human vision inspired action recognition using adaptive spatiotemporal sampling. *IEEE Trans. Image Process.*, 32: 5245–5256, 2023. doi: 10.1109/TIP.2023.3310661. URL https://doi.org/10.1109/TIP.2023.3310661.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In Aleksandra Faust, David Hsu, and Gerhard Neumann (eds.), *Conference on Robot Learning, 8-11 November 2021, London, UK*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1678–1690. PMLR, 2021. URL https://proceedings.mlr.press/v164/mandlekar22a.html.

Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. Recurrent models of visual attention. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 2204–2212, 2014. URL https://proceedings.neurips.cc/paper/2014/hash/09c6c3783b4a70054da74f2538ed47c6-Abstract.html.

Tharun Mohandoss, Sourav Pal, and Pabitra Mitra. Visual attention for behavioral cloning in autonomous driving. In Antanas Verikas, Dmitry P. Nikolaev, Petia Radeva, and Jianhong Zhou (eds.), *Eleventh International Conference on Machine Vision, ICMV 2018, Munich, Germany, 1-3 November 2018*, volume 11041 of *SPIE Proceedings*, pp. 110411E. SPIE, 2018. doi: 10.1117/12.2522915. URL https://doi.org/10.1117/12.2522915.

Athanasios Papadopoulos, Pawel Korus, and Nasir D. Memon. Hard-attention for scalable image classification. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 14694–14707, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/7b7916dd2de56297aa29cccb2bbf48d4-Abstract.html.

Adam Pardyl, Michal Wronka, Maciej Wolczyk, Kamil Adamczewski, Tomasz Trzcinski, and Bartosz Zielinski. Adaglimpse: Active visual exploration with arbitrary glimpse position and scale. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XXI*, volume 15079 of *Lecture Notes in Computer Science*, pp. 112–129. Springer, 2024. doi: 10.1007/978-3-031-72664-4_7. URL https://doi.org/10.1007/978-3-031-72664-4_7.

Samrudhdhi B. Rangrej and James J. Clark. A probabilistic hard attention model for sequentially observed scenes. In *32nd British Machine Vision Conference 2021, BMVC 2021, Online, November 22-25, 2021*. BMVA Press, 2021. URL https://www.bmvc2021-virtualconference.com/assets/papers/0251.pdf.

Samrudhdhi B. Rangrej, Chetan L. Srinidhi, and James J. Clark. Consistency driven sequential transformers attention model for partially observable scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA,*

USA, June 18-24, 2022, pp. 2508–2517. IEEE, 2022. doi: 10.1109/CVPR52688.2022.00255. URL https://doi.org/10.1109/CVPR52688.2022.00255.

Samrudhdhi B. Rangrej, Kevin J. Liang, Tal Hassner, and James J. Clark. Glitr: Glimpse transformers with spatiotemporal consistency for online action prediction. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023, Waikoloa, HI, USA, January 2-7, 2023*, pp. 3402–3412. IEEE, 2023. doi: 10.1109/WACV56688.2023.00341. URL https://doi.org/10.1109/WACV56688.2023.00341.

Akanksha Saran, Ruohan Zhang, Elaine Schaertl Short, and Scott Niekum. Efficiently guiding imitation learning agents with human gaze. In Frank Dignum, Alessio Lomuscio, Ulle Endriss, and Ann Nowé (eds.), *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021*, pp. 1109–1117. ACM, 2021. doi: 10.5555/3463952.3464081. URL https://www.ifaamas.org/Proceedings/aamas2021/pdfs/p1109.pdf.

Shikhar Sharma, Ryan Kiros, and Ruslan Salakhutdinov. Action recognition using visual attention. *CoRR*, abs/1511.04119, 2015. URL http://arxiv.org/abs/1511.04119.

Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S. Emer. Efficient processing of deep neural networks: A tutorial and survey. *Proc. IEEE*, 105(12):2295–2329, 2017. doi: 10.1109/JPROC.2017.2761740. URL https://doi.org/10.1109/JPROC.2017.2761740.

Chittesh Thavamani, Mengtian Li, Nicolas Cebron, and Deva Ramanan. FOVEA: foveated image magnification for autonomous navigation. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 15519–15528. IEEE, 2021. doi: 10.1109/ICCV48922.2021.01525. URL https://doi.org/10.1109/ICCV48922.2021.01525.

Brevin Tilmon and Sanjeev J. Koppal. Saccadecam: Adaptive visual attention for monocular depth sensing. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 5989–5998. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00595. URL https://doi.org/10.1109/ICCV48922.2021.00595.

Gang Wang, Wenmin Wang, Jingzhuo Wang, and Yaohua Bu. Better deep visual attention with reinforcement learning in action recognition. In *IEEE International Symposium on Circuits and Systems, ISCAS 2017, Baltimore, MD, USA, May 28-31, 2017*, pp. 1–4. IEEE, 2017. doi: 10.1109/ISCAS.2017.8050638. URL https://doi.org/10.1109/ISCAS.2017.8050638.

Yulin Wang, Kangchen Lv, Rui Huang, Shiji Song, Le Yang, and Gao Huang. Glance and focus: a dynamic approach to reducing spatial redundancy in image classification. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1963bd5135521d623f6c29e6b1174975-Abstract.html.

Yulin Wang, Zhaoxi Chen, Haojun Jiang, Shiji Song, Yizeng Han, and Gao Huang. Adaptive focus for efficient video recognition. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 16229–16238. IEEE, 2021. doi: 10.1109/ICCV48922.2021.01594. URL https://doi.org/10.1109/ICCV48922.2021.01594.

Yulin Wang, Yang Yue, Yuanze Lin, Haojun Jiang, Zihang Lai, Victor Kulikov, Nikita Orlov, Humphrey Shi, and Gao Huang. Adafocus V2: end-to-end training of spatial dynamic networks for video recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 20030–20040. IEEE, 2022a. doi: 10.1109/CVPR52688.2022.01943. URL https://doi.org/10.1109/CVPR52688.2022.01943.

Yulin Wang, Yang Yue, Xinhong Xu, Ali Hassani, Victor Kulikov, Nikita Orlov, Shiji Song, Humphrey Shi, and Gao Huang. Adafocusv3: On unified spatial-temporal dynamic video recognition. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part IV*, volume 13664 of *Lecture Notes in Computer Science*, pp. 226–243. Springer, 2022b. doi: 10.1007/978-3-031-19772-7_14. URL https://doi.org/10.1007/978-3-031-19772-7_14.

Yulin Wang, Haoji Zhang, Yang Yue, Shiji Song, Chao Deng, Junlan Feng, and Gao Huang. Uni-adafocus: Spatial-temporal dynamic computation for video recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(3):1782–1799, 2025. doi: 10.1109/TPAMI.2024.3514654. URL https://doi.org/10.1109/TPAMI.2024.3514654.

Howard Xiao, Jan Ackermann, Boyang Deng, and Gordon Wetzstein. Policy-based foveated imaging and perception. In Christopher P. Redmann, Justin Solomon, and Mirela Ben-Chen (eds.), *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference, SIGGRAPH Conference Papers 2026, Los Angeles, CA, USA, July 19-23, 2026*, pp. 87:1–87:11. ACM, 2026. doi: 10.1145/3799902.3811218. URL https://doi.org/10.1145/3799902.3811218.

Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 2678–2687. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.293. URL https://doi.org/10.1109/CVPR.2016.293.

Ruohan Zhang, Zhuode Liu, Luxin Zhang, Jake Alden Whritner, Karl S. Muller, Mary M. Hayhoe, and Dana H. Ballard. AGIL: learning attention from human for visuomotor tasks. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XI*, volume 11215 of *Lecture Notes in Computer Science*, pp. 692–707. Springer, 2018. doi: 10.1007/978-3-030-01252-6_41. URL https://doi.org/10.1007/978-3-030-01252-6_41.

Zheyu Zhuang, Ruiyu Wang, Nick Heppert, Johannes Fabian Hahn, Abhinav Valada, Florian T. Pokorny, and Danica Kragic. Attention from action, for action: Emergent visual bottlenecks for policy learning, 2026. URL https://arxiv.org/abs/2608.13422.

A Ablations

We organize the ablations along three axes: the downstream action-policy architecture, the sampling geometry, and the deployment crop width. Throughout this appendix, *fixed center* denotes a crop whose location never moves, whereas *fixed width* denotes GlimpseBC with learned, time-varying fixations but no field-of-view annealing. Annealed GlimpseBC also learns its fixations, but its source width shrinks during training.

A.1 Downstream Action-Policy Architecture

Table 7 repeats the fixed-center versus GlimpseBC comparison under both downstream policy families for the shared Push-T crop widths. Each center/GlimpseBC pair uses the same downstream policy and evaluation protocol. The RNN uses the regression objective, whereas ChiUNet uses flow matching, so the cross-family columns show that the result persists across policy architectures rather than providing a controlled ranking between the two architectures.

	RNN regression		ChiUNet flow matching	
Crop (px)	Fixed center	GlimpseBC	Fixed center	GlimpseBC
32	47.55	50.27	52.29	70.63
48	54.09	76.56	62.12	79.31
64	57.83	73.22	58.68	65.88

Table 7: Push-T normalized peak overlap (%) across downstream action-policy architectures. Comparisons are controlled within, but not across, each architecture family. ChiUNet entries use nine sampling steps.

A.2 Sampling Geometry

We also vary the geometry of the learned glimpse while keeping the RNN policy, training objective, output width, and evaluation protocol fixed. The uniform variant extracts the square crop used by GlimpseBC. The foveated variant instead uses a finite circular local view with a sharp center and a Gaussian-blurred periphery. Its source diameter equals the output width, the sharpness scale is 0.4 times the outer radius, and the peripheral blur has standard deviation 2 in output pixels. It adds neither trainable parameters nor a full-frame context branch.

(a) Rollout performance.

Crop (px)	Uniform	Foveated
24	45.72	47.86
32	50.27	56.18
48	76.56	72.81
64	73.22	73.91

(b) Reported model compute (uniform / foveated).

Crop (px)	Inference GFLOPs/input	Training TFLOPs/run
24	0.757 / 0.757	14,653.1 / 14,658.2
32	0.792 / 0.793	15,332.7 / 15,340.8
48	1.036 / 1.037	20,011.0 / 20,027.3
64	1.236 / 1.239	23,856.4 / 23,884.2

Table 8: Push-T sampling-geometry ablation for fixed-width GlimpseBC with the RNN regression policy. Overlap values are normalized peak overlap percentages. No field-of-view annealing is used.

Foveation improves the 24-, 32-, and 64-pixel results but not the 48-pixel result. We therefore treat geometry as an ablation rather than selecting one geometry using these rollout outcomes.

A.3 Visual Budget

The crop-width rows in Tables 7 and 8 vary the deployment visual budget while holding the corresponding architecture and geometry fixed. The tables below retain the complete task-level results, sampling-step results, and compute rather than averaging across tasks or crop widths.

A.3.1 Complete RNN Results and Compute

RoboMimic		**Push-T**		**LIBERO-Spatial**			
Task	Success (%)	Task	Overlap	Task	Success (%)	Task	Success (%)
Can	80	Push-T	79.01	0	62	5	90
Lift	99			1	28	6	74
Square	70			2	82	7	64
Tool Hang	10			3	78	8	44
				4	70	9	36

Table 9: Full-frame RNN baselines. RoboMimic and LIBERO report success rates (%), while Push-T reports normalized overlap as a percentage.

Task	Crop (px)	Performance (%)	Training TFLOPs	Inference GFLOPs
Can	32	48.00	12,981.5	0.675
Can	48	72.00	22,189.6	1.162
Can	64	82.00	29,669.1	1.563
Lift	32	100.00	17,305.8	0.900
Lift	48	100.00	26,513.9	1.388
Lift	64	99.00	33,993.4	1.788
Square	32	16.00	12,981.5	0.675
Square	48	52.00	22,189.6	1.162
Square	64	79.00	29,669.1	1.563
Tool Hang	32	19.00	12,981.5	0.675
Tool Hang	48	12.00	22,189.6	1.162
Tool Hang	64	8.00	29,669.1	1.563
Push-T	32	47.55	14,442.0	0.750
Push-T	48	54.09	19,043.1	0.994
Push-T	64	57.83	22,780.4	1.194

Table 10: Fixed center-crop RNN performance and compute. Training compute is TFLOPs per run; inference compute is GFLOPs per policy input.

Table 11: Full-frame ChiUNet evaluation results. Training FLOPs are reported in PFLOPs, while evaluation FLOPs are reported in TFLOPs. Evaluation FLOPs include all evaluated NFE settings.

		Success Rate			Mean Reward			Compute	
Benchmark	Task	NFE 1	NFE 3	NFE 9	NFE 1	NFE 3	NFE 9	Train PFLOPs	Eval TFLOPs
RoboMimic	Can	0.970	1.000	0.980	35.190	35.890	36.160	47.766	45.587
RoboMimic	Lift	1.000	1.000	1.000	44.460	44.340	45.170	47.766	45.587
RoboMimic	Square	0.900	0.860	0.910	28.330	25.590	28.440	47.766	45.587
Push-T	Push-T	0.609	0.622	0.590	15.908	14.247	14.071	28.093	22.816
LIBERO-Spatial	Task 0	0.820	0.720	0.720	0.820	0.720	0.720	92.134	14.752
LIBERO-Spatial	Task 1	0.680	0.860	0.740	0.680	0.860	0.740	92.134	17.107
LIBERO-Spatial	Task 2	1.000	0.940	0.980	1.000	0.940	0.980	92.134	11.600
LIBERO-Spatial	Task 3	0.780	0.920	0.920	0.780	0.920	0.920	92.134	12.853
LIBERO-Spatial	Task 4	0.940	0.980	0.940	0.940	0.980	0.940	92.134	15.121
LIBERO-Spatial	Task 5	0.680	0.920	0.880	0.680	0.920	0.880	92.134	14.716
LIBERO-Spatial	Task 6	0.820	0.860	0.880	0.820	0.860	0.880	92.134	14.746
LIBERO-Spatial	Task 7	0.840	0.980	0.980	0.840	0.980	0.980	92.134	15.648
LIBERO-Spatial	Task 8	0.740	0.780	0.720	0.740	0.780	0.720	92.134	17.895
LIBERO-Spatial	Task 9	0.660	0.820	0.760	0.660	0.820	0.760	92.134	18.464

A.3.2 Complete ChiUNet Results

Task	Crop (px)	Success or overlap (%; 9/3/1)	Training (TFLOPs)	Inference (GFLOPs/input; 9/3/1)
Can	32	67 / 65 / 63	8,697.0	1.687 / 0.761 / 0.452
Can	48	82 / 87 / 82	17,905.1	2.174 / 1.248 / 0.939
Can	64	96 / 97 / 92	25,384.6	2.575 / 1.649 / 1.340
Lift	32	100 / 100 / 100	8,697.0	1.687 / 0.761 / 0.452
Lift	48	100 / 100 / 100	17,905.1	2.174 / 1.248 / 0.939
Lift	64	100 / 100 / 100	25,384.6	2.575 / 1.649 / 1.340
Square	32	31 / 27 / 19	8,697.0	1.687 / 0.761 / 0.452
Square	48	73 / 72 / 72	17,905.1	2.174 / 1.248 / 0.939
Square	64	88 / 85 / 80	25,384.6	2.575 / 1.649 / 1.340
Tool Hang	32	40 / 16 / 19	8,697.0	1.687 / 0.761 / 0.452
Tool Hang	48	36 / 3 / 19	17,905.1	2.174 / 1.248 / 0.939
Tool Hang	64	32 / 10 / 12	25,384.6	2.575 / 1.649 / 1.340
Push-T	32	52.29 / 48.16 / 47.09	5,857.8	1.536 / 0.611 / 0.303
Push-T	48	62.12 / 63.00 / 64.87	10,458.8	1.780 / 0.855 / 0.547
Push-T	64	58.68 / 59.12 / 60.96	14,196.1	1.980 / 1.055 / 0.747

Table 12: Fixed center-crop results using the standard image encoder, a ChiUNet policy, and the flow-matching objective. Success and inference-compute entries are reported in the order 9/3/1 sampling steps.

Task	Patch (px)	Success or overlap (%; 9/3/1)	Training (TFLOPs)	Inference (GFLOPs/input; 9/3/1)
Can	32	99 / 99 / 98	9,228.6	1.707 / 0.781 / 0.472
Can	48	96 / 95 / 95	18,591.2	2.195 / 1.269 / 0.960
Can	64	98 / 99 / 99	26,287.1	2.596 / 1.669 / 1.361
Lift	32	100 / 100 / 100	9,228.6	1.707 / 0.781 / 0.472
Lift	48	100 / 100 / 100	18,591.2	2.195 / 1.269 / 0.960
Lift	64	100 / 100 / 100	26,287.1	2.596 / 1.669 / 1.361
Square	32	88 / 86 / 90	9,228.6	1.707 / 0.781 / 0.472
Square	48	91 / 86 / 77	18,591.2	2.195 / 1.269 / 0.960
Square	64	91 / 87 / 80	26,287.1	2.596 / 1.669 / 1.361
Tool Hang	32	62 / 12 / 14	9,228.6	1.707 / 0.781 / 0.472
Tool Hang	48	38 / 10 / 12	18,591.2	2.195 / 1.269 / 0.960
Tool Hang	64	44 / 6 / 34	26,287.1	2.596 / 1.669 / 1.361
Push-T	32	70.63 / 68.66 / 69.00	6,128.7	1.547 / 0.622 / 0.313
Push-T	48	79.31 / 70.92 / 71.41	10,806.9	1.790 / 0.866 / 0.557
Push-T	64	65.88 / 60.31 / 68.97	14,652.3	1.991 / 1.066 / 0.758

Table 13: Learned-fixation results using the standard GlimpseBC image encoder, a ChiUNet policy, and the flow-matching objective. Success and inference-compute entries are reported in the order 9/3/1 sampling steps.

A.3.3 LIBERO-Spatial RNN Results and Compute

Task	Full frame	GlimpseBC 32	GlimpseBC 48	GlimpseBC 64	GlimpseBC 96
0	62	46	88	84	82
1	28	0	66	56	74
2	82	50	12	72	82
3	78	40	54	84	80
4	70	2	18	30	44
5	90	66	10	84	60
6	74	0	32	76	70
7	64	0	38	26	0
8	44	72	74	68	90
9	36	12	0	6	14
Macro mean	62.8	28.8	39.2	58.6	59.6

Table 14: Full-frame and fixed-width GlimpseBC RNN success rates (%) on LIBERO-Spatial. Fixed-width GlimpseBC uses learned, time-varying fixations and does not anneal its field of view.

Deployment view	Reported training TFLOPs/run	Inference GFLOPs/input
Full frame (128×128)	96,416.9	5.117
32 px	14,719.09	0.756
48 px	24,081.75	1.244
64 px	31,777.59	1.645
96 px	60,208.42	3.126

Table 15: LIBERO-Spatial RNN model compute indexed by deployment width. For annealed runs, these values do not integrate the changing crop size over the training schedule.

B Offline Validation for Annealing Schedules

The main paper reports the action N-MSE used to select each initial crop width S_0. Tables 16–18 give all recorded validation metrics. The full-chunk metrics average error over the entire predicted action sequence, while the executed-prefix metrics use only the actions the policy executes before replanning. Both are computed in the normalized action space used for policy training. In each group, bold marks the S_0 with the lowest full-chunk N-MSE, which is the selection criterion. RoboMimic has only one initial width for each endpoint and therefore does not provide a schedule-selection comparison.

		Full chunk		Executed prefix	
S_T	S_0	N-MSE	N-MAE	N-MSE	N-MAE
24	48	0.025726	0.096664	0.012876	0.069852
	72	**0.023068**	0.090473	0.011708	0.066095
	96	0.024223	0.092157	0.012147	0.067143
32	48	0.023999	0.094776	0.012361	0.069459
	72	0.022810	0.090559	0.012285	0.067114
	96	**0.021688**	0.088753	0.011579	0.065998
48	72	0.022612	0.092080	0.012170	0.068621
	96	**0.021090**	0.089215	0.011541	0.066987
64	72	0.023463	0.095122	0.012779	0.070879
	96	**0.021434**	0.090141	0.011601	0.067261

Table 16: Push-T offline validation for annealed GlimpseBC. Bold marks the selected initial width S_0 for each deployment width S_T.

		Full chunk		Executed prefix	
Task	S_T	N-MSE	N-MAE	N-MSE	N-MAE
Can	32	0.024887	0.065051	0.025561	0.065499
	48	0.024821	0.063659	0.025038	0.064002
	64	0.025839	0.066260	0.025862	0.066431
Lift	32	0.048528	0.114139	0.038761	0.098115
	48	0.050567	0.116512	0.040541	0.100816
	64	0.052244	0.117231	0.042567	0.101862
Square	32	0.035815	0.080039	0.035492	0.080099
	48	0.033151	0.078847	0.032915	0.078822
	64	0.033490	0.079008	0.033177	0.078905
Tool Hang	32	0.021217	0.047045	0.021310	0.046584
	48	0.020472	0.044413	0.020390	0.043954
	64	0.021704	0.047164	0.021798	0.046542

Table 17: RoboMimic offline validation for annealed GlimpseBC. Every run starts at $S_0 = 84$ pixels, so each row is the only schedule for its task and deployment width S_T.

			Full chunk		Executed prefix	
Task	S_T	S_0	N-MSE	N-MAE	N-MSE	N-MAE
1	32	64	0.077804	0.152160	0.078626	0.152567
		96	**0.073074**	0.151650	0.074079	0.152216
		128	0.079789	0.156261	0.081040	0.156949
	48	**64**	**0.073814**	0.148486	0.074772	0.149052
		96	0.079631	0.155588	0.081251	0.156682
		128	0.077997	0.153858	0.079472	0.154706
	64	96	0.077329	0.154187	0.078554	0.155066
		128	**0.071642**	0.146305	0.073362	0.147158
	96	128†	0.083373	0.156885	0.085010	0.157822
2	32	64	0.070098	0.145359	0.071866	0.146429
		96	**0.066109**	0.143359	0.066755	0.143862
		128	0.066229	0.140709	0.067304	0.141225
	48	64	0.071335	0.145299	0.072455	0.145550
		96	**0.070074**	0.147963	0.071399	0.148961
		128	0.071409	0.147244	0.071927	0.147531
	64	**96**	**0.073061**	0.147580	0.074225	0.148310
		128	0.074320	0.149002	0.074505	0.148995
	96	128†	0.068073	0.143618	0.069113	0.144193
3	32	**64**	**0.066799**	0.146901	0.067318	0.146862
		96	0.073521	0.147611	0.075215	0.148398
		128	0.078366	0.152982	0.080546	0.153992
	48	64	0.076817	0.153484	0.079131	0.154802
		96	0.076148	0.147904	0.078072	0.148565
		128	**0.075652**	0.150184	0.077499	0.150995
	64	96	0.068394	0.143532	0.070177	0.144610
		128	**0.067723**	0.142342	0.069789	0.143598
	96	128†	0.069993	0.143636	0.071658	0.144229

Table 18: LIBERO-Spatial (tasks 1–3) offline validation for annealed GlimpseBC. Selection is performed independently for each task and deployment width S_T; bold marks the selected S_0. †Sole candidate; no selection was made.

C Selected Push-T Annealing Results

Model	View (px)	Overlap	Infer GF/input	Reported train TF
Baseline	84×84 crop	0.7901	1.787	33963.6
GlimpseBC, fixed width	24	0.4572	0.757	14653.1
GlimpseBC, fixed width	32	0.5027	0.792	15332.7
GlimpseBC, fixed width	48	0.7656	1.036	20011.0
GlimpseBC, fixed width	64	0.7322	1.236	23856.4
GlimpseBC, annealed	$72 \rightarrow 24$	0.5759	0.757	14653.1
GlimpseBC, annealed	$96 \rightarrow 32$	0.6259	0.792	15332.7
GlimpseBC, annealed	$96 \rightarrow 48$	0.7853	1.036	20011.0
GlimpseBC, annealed	$96 \rightarrow 64$	0.7456	1.236	23856.4

Table 19: Selected Push-T annealing results. Training seed 0 and evaluation seed 10,000. Overlap is normalized Push-T score. The reported compute is indexed by deployment width and should not be interpreted as schedule-integrated training cost for the annealed runs.

D GliTr Results

We use the official GliTr implementation and make only the changes needed to apply it to behavioral cloning: expert actions replace video-class labels as the prediction target, and the input and output interfaces are adjusted to each benchmark. We otherwise retain GliTr's teacher–student design, sequential glimpse selection, spatial and temporal consistency objectives, and training pipeline. The teacher observes the full frame, while the student selects each new crop from its previous glimpses. We evaluate the final checkpoint at crop widths of 32, 48, and 64 pixels. Because GliTr retains its own downstream policy, these results are reported separately from the controlled architecture, geometry, and crop-width ablations above.

Task	Teacher 84×84	Student $p2$ 32×32	Student $p3$ 48×48	Student $p4$ 64×64
Can	93.0	0.0	72.0	85.0
Lift	99.0	2.0	3.0	6.0
Square	79.0	11.0	41.0	68.0

Table 20: GliTr performance on the RoboMimic tasks. Values report final task success rate (%). The teachers observe the full 84×84 image, while the $p2$, $p3$, and $p4$ students use 32×32, 48×48, and 64×64 inputs, respectively.

Task	Teacher	Student (32 px)	Student (48 px)	Student (64 px)
Push-T	77.1%	11.7%	24.7%	48.6%

Table 21: GliTr results on Push-T. The teacher observes the full frame; students use crops of 32, 48, or 64 pixels. Performance is normalized overlap at training step 199999, expressed as a percentage.

Task	Teacher	Student (32 px)	Student (48 px)	Student (64 px)
0	62.0%	16.0%	38.0%	48.0%
1	96.0%	58.0%	52.0%	86.0%
2	100.0%	58.0%	66.0%	64.0%
3	74.0%	30.0%	42.0%	40.0%
4	90.0%	0.0%	10.0%	38.0%
5	60.0%	36.0%	54.0%	60.0%
6	94.0%	–	34.0%	–
7	94.0%	38.0%	50.0%	58.0%
8	74.0%	8.0%	62.0%	68.0%
9	38.0%	10.0%	12.0%	26.0%
Macro mean	78.2%	28.2%	42.0%	54.2%

Table 22: GliTr results on the ten LIBERO-Spatial tasks. The teacher observes the full frame; students use crops of 32, 48, or 64 pixels. Values are episode success rates at training step 199999. Dashes denote unavailable results; macro means average completed tasks only.