CCPL: CONTINUOUS CAUSAL PREDICTIVE LEARNING FOR SPATIALLY QUERYABLE VISUAL REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce Continuous Causal Predictive Learning (CCPL), a self-supervised framework that accumulates a causal state from local glimpses and predicts a continuous-valued next-glimpse latent conditioned on real-valued relative pan and zoom through a conditional flow matching objective. Unlike masked or autoregressive image models, whose predictions are specified by a mask position on a fixed patch grid or a fixed scan order, CCPL conditions each prediction on a continuous relative move, and hence the frozen model can be queried at independently specified positions and scales. After pretraining on ImageNet, query error decreases as a prescribed random-walk history grows and increases when recurrent memory is reset or query-move conditioning is shuffled. These controls provide operational evidence that the causal state accumulates information useful for spatial prediction across the sequence. We then freeze the observer and predictor and train an explorer with a category-label-free terminal reward given by normalized query error from the final feature. At a fixed budget of 1024 glimpses, learned scans reach the lowest terminal query error among all evaluated scans on both CCPL models and raise final-feature linear-probe top-1 over matched random walks by 1.18 and 1.34 points. However, recognition-leading lattice scans have higher query MSE than random walks. On the learned-scan's final and mean features, two-layer probes raise top-1 by 9 to 11 points over the corresponding linear probes, suggesting that class information in CCPL features is better accessed nonlinearly. Thus, a frozen predictive model can supply a category-label-free acquisition signal, but reconstruction and recognition favor different acquisition strategies and require separate evaluation.

1 INTRODUCTION

To predict visual content beyond its current observation, a model must retain information from what it has already seen. Prediction is therefore an appealing training signal: it requires no semantic labels, yet it pressures the learned representation to retain visual information. Context encoders and masked autoencoders learn transferable features in this way (Pathak et al., 2016; He et al., 2022). We treat this implication as two empirical questions rather than premises: does the representation support accurate reconstruction at queried poses, and is the retained information *linearly accessible*? We therefore measure reconstruction and frozen-feature recognition separately.

A predictive objective does not by itself make its representation spatially queryable. As illustrated in Figure 1, predictive objectives differ in the mechanism used to specify their prediction targets. Masked autoencoders identify targets by mask position, while AIM binds prediction to a fixed raster order and CPC to future positions in an observed sequence (He et al., 2022; El-Nouby et al., 2024; van den Oord et al., 2018). The Generative Query Network instead renders from a specified camera pose, showing that a predictive representation can expose a spatial-query interface (Eslami et al., 2018). We study that interface under partial sequential observation: an observer collects a causal sequence of local glimpses linked by real-valued relative movement and scale. Under a chosen observation budget, which views it acquires determines what can later be reconstructed or recognized.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

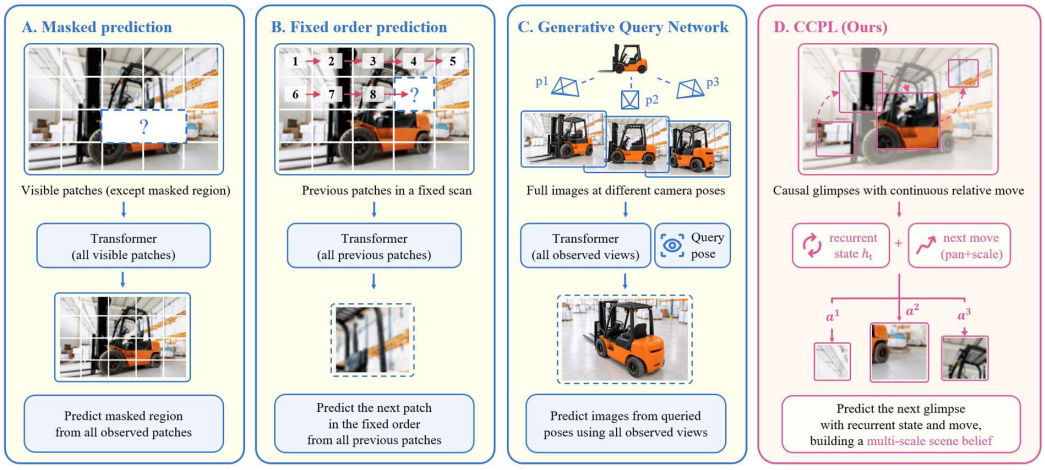

Figure 1: **Comparison of previous methods and the CCPL framework.** Prior objectives specify the target through a masked region, a fixed scan order, or a queried full-view pose [(a)∼(c)]. In contrast, CCPL accumulates a causal history of local glimpses and predicts the next glimpse from a continuous relative move [(d)]. Therefore, one recurrent state can be queried at previously unobserved positions and scales.

We call the resulting learning problem *Continuous Causal Predictive Learning* (CCPL): predicting a continuous-valued observation latent at a continuously specified relative pose from a causal state built from past glimpses. We instantiate its causal state with MambaEye (Choi et al., 2026), a size-agnostic vision encoder that updates a fixed-size recurrent state once per glimpse, so a single model is pretrained on images of varying size at a cost linear in the number of glimpses. The CCPL objective does not itself require recurrence. We extend MambaEye's relative planar pan with a log-scale channel for zoom. A frozen non-quantized patch codec maps each patch to a continuous-valued latent target, rather than a finite codebook index as in codebook-based image generation (Esser et al., 2021). Conditioned on the causal history and a real-valued relative pan and zoom, a flow-matching predictor models the next patch latent. Thus, "continuous" describes both the visual target and the query condition, while "causal" means that no future observation reaches the state that predicts it. During CCPL pretraining, observation trajectories follow a random walk rather than a fixed scan path or the learned explorer.

As an application, this interface makes the trained model reusable for active perception: because the predictor can be queried at independently specified poses (Figure 1-(d)), its error scores how much of the scene the current history already explains. We freeze the codec, observer and predictor, and train a separate explorer on this error after its final glimpse, normalized by the corresponding error of a fixed random scan. The targets come from the same ImageNet training set used for pretraining, without class label. We then freeze the explorer and fit linear heads on the features its observations induce, asking whether observations selected solely to reduce reconstruction error also make class information linearly accessible.

Contributions.

- **A continuous query interface for causal visual prediction.** We formulate CCPL, which predicts a continuous-valued local observation latent from a causal glimpse history and a continuous relative move. This turns a partial observation state into a spatially queryable predictive representation, rather than binding targets to a mask or a fixed scan order.
- **Operational diagnostics of the queryable state.** On a frozen CCPL model, spatial-query error improves with longer histories and worsens when recurrent memory or query-move conditioning is perturbed, directly testing the roles of causal history and relative movement.
- **Reconstruction-reward acquisition.** We reuse the frozen predictor as a spatial-query scorer whose normalized terminal error provides the entire category-label-free reward for a separate explorer, without jointly training the representation for recognition.

Method	Context	Processing	Predicts	Conditioned on
SimCLR / BYOL / DINO	augmented views	parallel	latent	—
MAE	visible patches	parallel	pixel	grid position
I-JEPA	visible patches	parallel	latent	grid position
CPC	past patches	causal, fixed scan	latent	scan step
AIM	past patches	causal, fixed scan	pixel	scan step
GQN	posed views	parallel	pixel	camera pose
CCPL (Ours)	past glimpses	causal, random walk	latent	relative move

Table 1: **Comparison of CCPL and related predictive objectives.**

- **A divergence between reconstruction and recognition.** Learned scans improve terminal query MSE and final-feature accuracy over matched random walks, whereas recognition-leading lattice scans have higher query MSE, empirically showing that reconstruction and recognition favor different acquisition strategies.

2 Related Work

Predictive self-supervised learning. Predicting withheld content is an established source of transferable features, and CCPL belongs to this family. Contrastive predictive coding (CPC) predicts future latents from an autoregressive context (van den Oord et al., 2018), and autoregressive image models (AIM) extend this idea to patch sequences under a fixed raster order (El-Nouby et al., 2024) over the patch-token interface of ViT (Dosovitskiy et al., 2021). Masked autoencoders (MAE) reconstruct missing pixels from sparse visible context (He et al., 2022), while I-JEPA predicts target *embeddings* instead of pixels (Assran et al., 2023). CCPL likewise pretrains on random-walk trajectories, but specifies each prediction by a continuous relative move rather than a mask position or a fixed scan order. Unlike I-JEPA, CCPL models frozen codec latents with a generative flow objective rather than predicting learned target embeddings. Table 1 compares these supervision geometries. Reported frozen-feature results across these methods are not directly comparable because their readout protocols differ.

Queryable representations and reusable predictors. Generative Query Network (GQN) builds a scene representation from observed view and viewpoint pairs, and renders a view at a queried pose not among the observations (Eslami et al., 2018). This serves as the closest precedent for CCPL's spatial-query readout. V-JEPA 2 (Assran et al., 2025) freezes a predictively pretrained video encoder and trains an action-conditioned predictor on top for image-goal planning, the same freeze-then-reuse pattern CCPL applies to glimpse acquisition. Our observer builds on MambaEye (Choi et al., 2026), a size-agnostic causal glimpse encoder with relative planar pose embeddings with a log-scale channel added, and replaces its recognition curriculum with relative move conditioned next-glimpse prediction. Flow matching (Lipman et al., 2023) and Gated DeltaNet-2 (Hatamizadeh et al., 2026) are used as existing components.

Choosing observations. Active perception studies how an observer selects where and at what scale to look (Bajcsy, 1988). Recurrent attention models process partial observations sequentially, typically for a supervised recognition reward (Mnih et al., 2014). A closer line treats exploration as *active observation completion*. In the look-around setting, Jayaraman & Grauman (2018) reward an agent for reconstructing unobserved views under a limited view budget without manual labels, then transfer the policy to recognition, and Ramakrishnan et al. (2019) extend this to glimpse sequences. Seifi & Tuytelaars (2019) drive exploration from spatial reconstruction error and classify scenes from the views collected, and Seifi et al. (2021) select glimpses with self-attention trained through downstream task losses rather than reinforcement learning. These approaches use discrete viewgrids, but AdaGlimpse (Pardyl et al., 2024) instead learns *continuous* glimpse position and scale with a task-loss reward, jointly training the encoder with the agent. Thus, label-free completion rewards, policy transfer, and continuous pan/zoom actions are not individually new in CCPL. Our focus is narrower: a *separately pretrained* causal next-glimpse predictor is frozen as an independent spatial-query scorer, and the resulting frozen observer and explorer are evaluated through linear recognition.

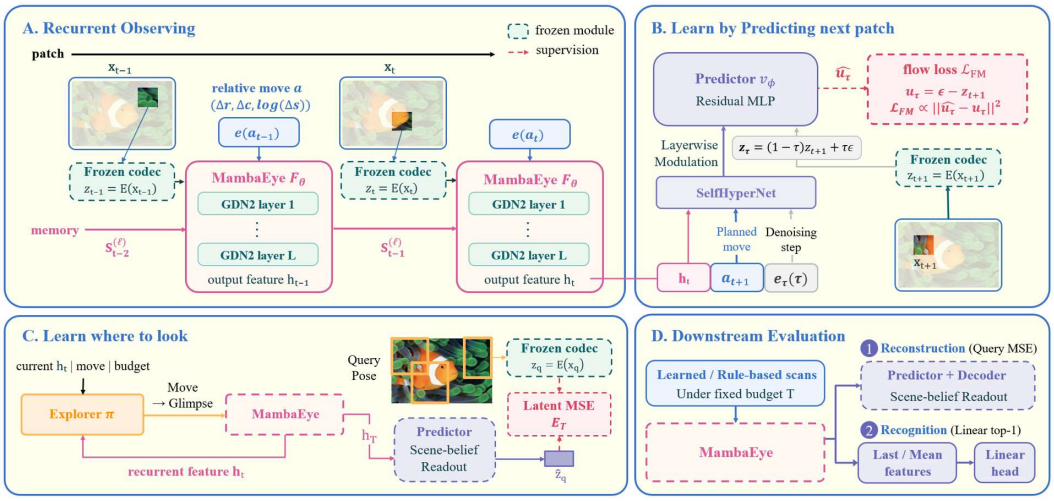

Figure 2: **CCPL: observe, predict, acquire, evaluate.** (A) MambaEye integrates encoded glimpses and continuous relative move. (B) A flow-matching predictor predicts the next glimpse latent from history and planned move. (C) With frozen codec, observer and predictor, an explorer minimizes normalized terminal latent query MSE. (D) The frozen system is evaluated through spatial-query error and final- and mean-feature linear probes.

Because AdaGlimpse trains its encoder with task-specific supervision, whereas our recognition heads read a frozen observer, its published results are not directly comparable to our protocol.

3 Method

We first formalize the common observation interface. CCPL then proceeds in three stages (Figure 2): predictive pretraining, explorer training against a frozen predictive scorer, and frozen readout. We describe the learning setup for each stage below. Further implementation details such as trajectory sampling geometry, the recurrent update, and the predictor growth recipe are in Appendices D, E, and F.

3.1 Observation interface and causal state

Let I be a scene of arbitrary size, extended by reflection to an infinite canvas, and $p_t = (r_t, c_t, s_t)$ the sensor pose at step t, denoting the row, column, and scale. A glimpse operator extracts a local observation $x_t = \mathcal{G}(I, p_t) \in \mathbb{R}^{3\times P\times P}$ by ROI alignment, with P=16. Movement between consecutive poses is defined as the three-dimensional continuous action

$$a_{t+1} = \left(\frac{r_{t+1}-r_t}{s_t P},\ \frac{c_{t+1}-c_t}{s_t P},\ \log\frac{s_{t+1}}{s_t}\right). \tag{1}$$

The first two coordinates are MambaEye's relative pan (Choi et al., 2026) in units of the current glimpse width. CCPL augments this action with a third coordinate, representing a relative log-scale change. The stored third coordinate is the *executed* increment after any boundary fold, not the unconstrained proposal. A sinusoidal embedding $e(a_t)$ is shared by the observer and predictor.

A frozen, non-quantized patch VAE E (Kingma & Welling, 2014) maps each glimpse to its deterministic posterior-mean latent $z_t = E(x_t) \in \mathbb{R}^{d_z}$, d_z=128. No codebook lookup or discrete tokenization is applied. The codec is frozen throughout CCPL and explorer training (architecture and recorded training settings in Appendix C). The observer F_θ is a MambaEye backbone whose mixing layers are Gated DeltaNet-2 (Hatamizadeh et al., 2026). Each layer ℓ carries a matrix-valued memory $S_t^{(\ell)}$, and the encoder maps previous memories, the current latent, and the executed move to updated memories and an *output feature* h_t:

$$\left(h_t,\ \{S_t^{(\ell)}\}_\ell\right) = F_\theta\left(\{S_{t-1}^{(\ell)}\}_\ell,\ z_t,\ e(a_t)\right), \qquad h_t \ \perp\ x_{>t} \ \middle|\ x_{\le t},\ a_{\le t}. \tag{2}$$

Equation 2 distinguishes the internal layer memories $S_t^{(\ell)}$ from the output feature h_t consumed by the predictor and the explorer policy. The observer computes h_t using only glimpses and actions available up to step t.

3.2 Relative-move-conditioned predictive objective

We train the observer F_θ and predictor v_ϕ jointly to predict the latent of the next glimpse. Given the observer feature h_t and the planned move a_{t+1}, the predictor is trained with a conditional flow-matching objective (Lipman et al., 2023). The continuous target is modeled directly through a flow path: we sample $\epsilon \sim \mathcal{N}(0, I)$ and $\tau \in (0, 1)$, and set $z_\tau = (1-\tau)z_{t+1} + \tau\epsilon$ and $u_\tau = \epsilon - z_{t+1}$. The objective is

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{t,\tau,\epsilon} \frac{1}{d_z} \big\| v_\phi\big(z_\tau,\, \tau,\, h_t,\, e(a_{t+1})\big) - u_\tau \big\|_2^2. \tag{3}$$

The loss updates both F_θ and v_ϕ, while the glimpse codec E remains frozen. Gradients through h_t train the observer to produce features useful for predicting the next glimpse given a planned move. The predictor conditions on h_t, the movement embedding, and the flow timestep through SelfHyper (Appendix D.1).

Because the observer is trained only through the predictor, a predictor too small to model the next glimpse would limit what the observer is pushed to encode. We therefore grow the predictor during pretraining, in the spirit of progressively grown generators (Karras et al., 2018): training starts from $\phi_{4,256}$ (residual MLP depth 4, SelfHyper rank 256) and reaches $\phi_{16,1280}$ in three function-preserving morphs (Chen et al., 2016), each starting from the function already learned (Appendix F). Validation flow-matching loss falls after every step on all three observers (from 1.127 to 1.060 on 1024w24d), although each step also adds training epochs, so the gain is not a pure capacity effect.

Because pretraining samples relative moves from a random walk and feeds each move to the predictor, the predictor learns to generalize across a distribution of sampled moves rather than a single fixed scan order. Given a single h_t and many relative-move queries, it predicts what the scene looks like at each queried pose from the observation history alone. Section 3.3 uses this to query the frozen model at independently sampled positions and scales.

3.3 The frozen predictor as a spatial-query scorer

After pretraining, we freeze codec, observer, and predictor. Given a history feature h at pose p, a relative move to an *independently sampled* query pose q is encoded as in Equation 1. Starting from Gaussian noise, the predictor generates an estimate $\bar{z}(h, p, q, \xi)$ of the latent glimpse at q, where ξ denotes the sampling noise. The query pose is sampled independently of the observer's trajectory. For a set of query poses Q, stratified with equal count per scale in $\{0.5, 1, 2, 4\}$, the *spatial-query error* compares the predictions with the latents of the corresponding glimpses from the full image:

$$\mathcal{E}(h, p; Q, \xi) = \frac{1}{|Q|\, d_z} \sum_{q \in Q} \big\| \bar{z}(h, p, q, \xi) - E\big(\mathcal{G}(I, q)\big) \big\|_2^2. \tag{4}$$

We refer to the scene information retained in h as the model's *scene belief*. The spatial-query error measures how accurately this belief predicts glimpses at independently sampled poses. We use the latent-space error to train the explorer (Section 3.4) and report decoded pixel-space error alongside it.

Exploration geometry. The explorer inherits the pretraining geometry: reflect padding, and scale bounds reflected at the same sensor-derived walls (Appendix E). To place all scanning methods on a common domain, each source image is cropped and resized onto a fixed 256×256 canvas, using random resized crops and horizontal flips during training and a deterministic resize and center crop for validation and feature extraction. Every episode starts at the canvas center with $s{=}1$, and a learned move may pan at most four glimpse widths and change log-scale by at most 0.35.

3.4 Category-label-free explorer

The spatial-query error measures how well the observed history predicts glimpses elsewhere in the scene. We use this signal to train an explorer that chooses where to observe next. An episode has a

budget of $T = 1024$, including the initial glimpse. The explorer steers the pretraining random walk rather than replacing it. It chooses one of every eight moves, those landing at glimpses $8, 16, \ldots, 1024$, while the other moves follow the pretraining random walk. The frozen observer reads every glimpse and retains its state throughout the episode. We evaluate the spatial-query error only at the end of the episode and define the terminal reward as

$$R = -\frac{\mathcal{E}(h_T, p_T; Q, \xi)}{\bar{\mathcal{E}}_T^{\mathrm{rw}}}, \tag{5}$$

where $\bar{\mathcal{E}}_T^{\mathrm{rw}}$ is the mean error the pretraining random walk reaches over the same budget on 256 fixed training images, computed once before training.

We optimize the explorer for the reward in Equation 5 using REINFORCE (Williams, 1992) with a leave-one-out baseline (Ahmadian et al., 2024) and an entropy bonus. The recurrent policy selects pan and log-zoom from the frozen observer feature h_t, the displacement since its previous decision, and the remaining budget. When selecting movements, the policy observes neither the query targets nor per-step reconstruction feedback. The full training image is used to compute the terminal reward, but is not available to the policy when it selects movements. Architecture and optimization details are given in Appendix A.

3.5 Frozen readout

We freeze the codec, observer, predictor, and explorer policy for evaluation. To evaluate scene prediction, we measure spatial-query error using a fresh query set Q and prediction noise ξ. To evaluate recognition, we record the observer features produced by each scan and fit linear classifiers on either the final feature h_T or the mean feature $T^{-1}\sum_{t=1}^{T} h_t$. Features are normalized beforehand, and each scanning method has its own classifier trained under an identical protocol. Category labels are used to fit classifiers and to select the explorer checkpoint on a development split (Section 4.1). For learned scans, features are extracted with sampled policy actions under a recorded seed.

4 Experiments

Evaluation separates spatial-query error (Section 4.2) from frozen-feature recognition under different acquisition strategies (Section 4.3). Recognition compares the learned scan, matched random walk, and nine lattice scanning patterns at the same 1024-glimpse budget. Direct query-capability tests evaluate the frozen 1024w24d CCPL model; matched terminal query MSE is reported alongside recognition on both evaluated observers.

4.1 Protocol

Pretraining. Pretraining uses ImageNet-1k (Russakovsky et al., 2015), 1024-glimpse sequences, 16×16 glimpses, and a frozen image patch codec with 128-D latents. Each image is randomly cropped, horizontally flipped, and resized with its aspect ratio kept so that its long side is drawn uniformly from 256 to 1024 pixels. All runs use a reflect-padded pretraining random walk with randomized pan scale and log-scale Brownian motion under physical scale boundaries. Geometry and optimization are detailed in Appendix A. The three configurations are 256w6d, 512w12d and 1024w24d (hidden width and depth), where width and depth vary together. The reported acquisition comparisons use the final 512w12d and 1024w24d observers.

Comparison scope. An *acquisition effect* compares scanning strategies within the same frozen CCPL model at matched canvas and budget. An isolated *objective effect* would require retraining alternative objectives under controlled architectures and observations, which we do not conduct in this work. External SSL systems are not benchmarked here, and published accuracies under other protocols are not matched baselines.

Common recognition protocol. Each scanning pattern yields one 1024-glimpse trajectory on a 256×256 crop. Separate linear heads read its final and mean features under the same fitting recipe (Section 3.5). All nine lattice patterns are reported, with numerical settings and motion/coverage

Condition	History length	Query MSE ↓ Latent	Pixel	Unobs. pixel	N_q
History length (all query scales)					
Full history	64	0.995	0.0590	0.0670	3748
Full history	128	0.952	0.0496	0.0622	3748
Full history	256	0.902	0.0403	0.0583	3748
Full history	512	<u>0.844</u>	<u>0.0321</u>	<u>0.0572</u>	3748
Full history	1024	**0.804**	**0.0273**	**0.0511**	3748
Conditioning controls					
Final-glimpse reset	1024	**1.117**	**0.0830**	**0.0900**	3748
Query-move shuffle (all)	1024	<u>1.203</u>	<u>0.0912</u>	<u>0.0935</u>	3748
Query-move shuffle (within scale)	1024	<u>1.203</u>	0.0931	0.0946	3748
Fixed-feature scale sweep					
Query scale 0.5	1024	**0.555**	0.0288	**0.0473**	2614
Query scale 1	1024	<u>0.757</u>	<u>0.0270</u>	<u>0.0555</u>	885
Query scale 2	1024	0.895	**0.0254**	0.0721	202
Query scale 4	1024	1.011	0.0282	0.0898	47

Table 2: **Spatial-query MSE of the frozen 1024w24d CCPL model under prescribed random-walk histories.** History lengths share prefixes, queries and noise; controls perturb the terminal readout. "Unobs. pixel" is pixel MSE on query footprints with no source-pixel overlap with the full history. N_q denotes the number of such queries. Within each block, bold and underline mark the best and second best per column. Evaluation details are in Appendix H.6.

differences disclosed below. Accuracies and effect estimates are displayed to two decimal places, and latent and pixel query MSEs to three and four. Differences and relative changes are computed before display rounding.

Matched acquisition comparison. Each comparison freezes the same codec, observer and predictor, starts from the center of a reflected 256×256 crop at $s{=}1$, and uses $T{=}1024$. The learned-scan and matched-random-walk episodes share all prescribed random-walk increments at non-policy moves. Linear heads are fit on $1{,}281{,}167$ ImageNet-1k training images and evaluated on $50{,}000$ validation images, one deterministic center crop each. Reconstruction uses the same images, final features and poses, with queries and generation noise paired by image. Explorer checkpoints are selected by development final-feature top-1 (cycle 900 for 512w12d and 1200 for 1024w24d) before official validation. Category labels enter at checkpoint selection and linear head fitting. We use one pretraining and one explorer seed per observer. Paired 95% image-bootstrap intervals use $10{,}000$ resamples and quantify variation over evaluation images. (Appendix H.4).

4.2 What can the frozen model predict from partial history?

The pretrained 1024w24d CCPL model supports spatial-query prediction from partial history. Table 2 evaluates prefixes of the same recorded pretraining random walk on all $50{,}000$ ImageNet validation images, using the 256×256 center crops above. Each image has 16 queries at each scale $\{0.5, 1, 2, 4\}$, with query poses and generation noise shared across conditions. Predictions average two samples after eight Euler steps. Appendix H.6 gives scoring and uncertainty details.

All-query pixel MSE decreases monotonically from 0.0590 at 64 glimpses to 0.0273 at 1024, while latent MSE falls from 0.995 to 0.804. Resetting memory and processing only the recorded last glimpse raises pixel MSE to 0.0830. Shuffling query-move conditioning while holding targets and noise fixed raises it to 0.0912; shuffling only within each scale raises it to 0.0931. These perturbations support dependence on causal history and spatial conditioning.

On query footprints without overlapping with the observation history, pixel MSE also falls from 0.0670 to 0.0511 across the same history lengths. This subset is small: 3748 queries across 2598 images, out of 3.2 million queries overall. Only 47 queries from 36 images remain at scale 4.The scale sweep reuses one terminal feature per image, its targets change with scale.beyond the pretraining

	512w12d				1024w24d						
	Linear ↑		MSE ↓		Linear ↑			Two-layer ↑		MSE ↓	
Scan	Final	Mean	Latent	Pixel	Final	Mean	Concat	Final	Mean	Latent	Pixel
Random walk	16.54	26.70	0.872	0.0353	18.32	26.34	26.70	27.06	37.71	0.804	0.0274
Learned	17.73	23.99	**0.845**	**0.0294**	19.66	25.98	26.56	28.63	37.19	**0.777**	**0.0228**
Shuffled grid †	18.21	**29.58**	0.910	0.0400	20.36	**29.16**	29.54	30.32	**39.94**	0.926	0.0416
Diagonal †	14.01	26.51	1.004	0.0690	16.49	27.20	29.05	23.93	37.70	1.031	0.0697
Spiral †	22.45	25.94	0.989	0.0610	**22.60**	25.31	28.81	**32.65**	34.88	0.963	0.0483
Snake C †	16.25	27.34	0.978	0.0639	18.42	27.45	29.30	26.04	37.01	0.998	0.0599
Snake H †	16.92	26.86	0.994	0.0672	20.37	27.98	**30.35**	28.83	38.09	1.025	0.0686
Raster C †	15.91	25.92	0.986	0.0657	18.33	26.97	28.81	26.11	36.74	1.004	0.0631
Raster H †	16.18	26.00	1.005	0.0707	18.34	26.30	28.92	26.56	36.16	1.018	0.0685
Hilbert †	15.75	27.68	0.991	0.0666	19.31	26.12	28.47	27.43	36.15	1.013	0.0626
Golden †	**23.66**	28.48	0.994	0.0564	20.82	27.87	29.15	29.92	37.83	1.018	0.0611

Table 3: **ImageNet top-1 (%) and terminal query MSE at T=1024 glimpses.** † marks a lattice scan; bold and underline mark the best and second best per column. Two-layer and Concat (linear, final + mean) heads are tuned on held-out training images (Appendix H.2). Learned explorer's scan beats all other scans in reconstruction MSE, but not in linear classification (Top-1).

scale range. Supplementary Figure 6 shows the same kind of multiscale readout from the 1024w24d explorer after 1024 glimpses, with the ground-truth image as the reference row.

4.3 Frozen-feature recognition at a fixed observation budget

Table 3 shows a readout-specific benefit of learned acquisition. With the observer frozen, reconstruction-reward explorers raise final-feature top-1 from 16.54% to 17.73% on 512w12d and from 18.32% to 19.66% on 1024w24d, gains of 1.18 and 1.34 percentage points over each observer's matched random walk. The mean feature, a secondary readout that the explorer's reward does not involve, falls by 2.70 and 0.36 points; the latter's 95% interval includes zero. A two-layer head on the same cached 1024w24d features keeps the final-feature gain, with the mean feature 0.52 points below the random walk, and raises both readouts by roughly 9 to 11 points, whereas a linear head tuned the same way on the concatenated features (Concat) does not, and ranks the learned scan last. CCPL features are trained for next-glimpse prediction, and nothing in the objective (Equation 3) asks them to be linearly separable by class; as reported for masked autoencoders (He et al., 2022), class information is present but less linearly separable, so linear probes may understate it.

Lattice scans set a stronger comparison. The lattice scans (Appendix H.1) hold scale fixed on a regular lattice, so their movement and coverage differ from the policy's; they are practical budget-matched baselines, not an isolated ordering ablation. On 1024w24d, Spiral scanning reaches 22.60% final-feature top-1 and the Shuffled grid 29.16% mean-feature top-1, 2.93 and 3.18 points above the learned scan; on 512w12d, Golden, the golden-ratio scan of MambaEye (Choi et al., 2026), gives the highest final-feature accuracy, 23.66%. A gain over the matched random walk therefore does not establish that learning selects the best scan. Under the two-layer head, the learned scan's mean-feature deficit narrows (it ranks 6th of 11 scans rather than 10th), but the lattice scans' final-feature lead remains. Appendix H.2 details the two-layer and Concat heads.

Reconstruction and evaluation cost. On the same 50,000 validation images, learned scanning lowers terminal latent query MSE from 0.872 to 0.845 on 512w12d and from 0.804 to 0.777 on 1024w24d, and pixel query MSE from 0.0353 to 0.0294 and from 0.0274 to 0.0228, relative to matched random walks; all four paired image-bootstrap intervals exclude zero (Appendix H.4). Every lattice scan has higher reconstruction error than its random-walk reference, although several recognize better under both heads, so reconstruction and recognition rank scans differently and the divergence is not an artifact of the linear readout. These errors concern queried patches; neither lower MSE nor higher linear-probe accuracy establishes total information sufficiency. Measured latency and peak memory are reported in Appendix Table 9: the learned scan's observation step is about 9 to 11 times slower than the random walk's, because the policy runs every eight glimpses.

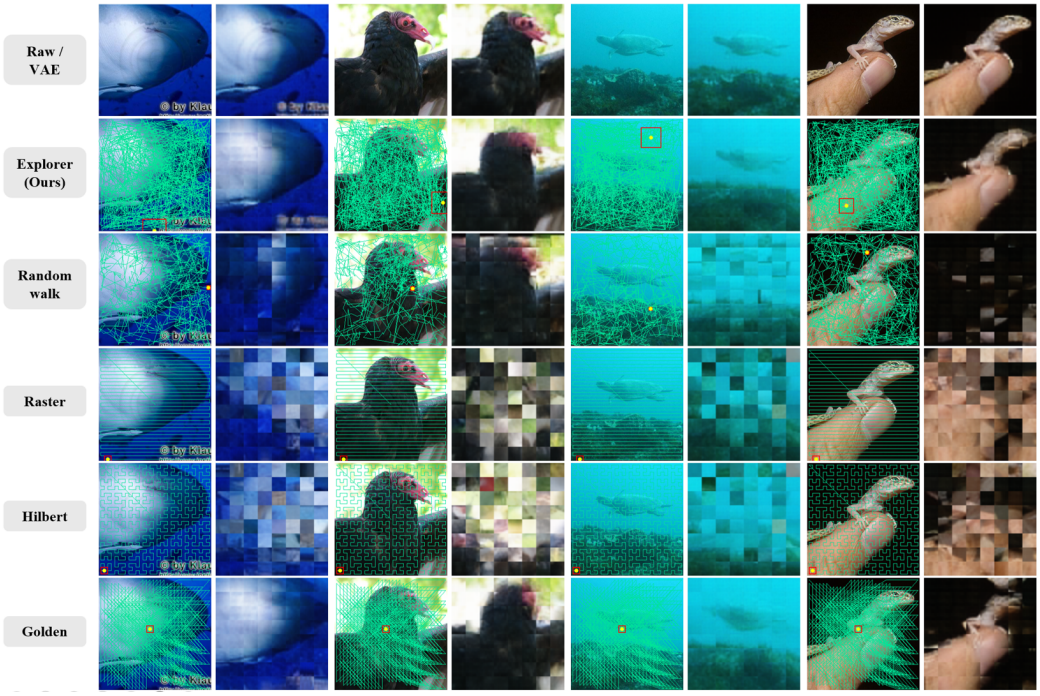

Figure 3: **Reconstruction under different scanning patterns.** Each row is one scan. The top row shows the raw image beside the codec reconstruction. The other rows are the explorer, a matched random walk, raster, Hilbert, and Golden: for each example the trajectory overlay is beside the reconstruction from that scan.

5 Conclusion, Limitations, and Future Work

CCPL learns visual representations by predicting continuous patch latents from local-glimpse histories conditioned on relative movement. This makes the frozen model spatially queryable and allows its query error to serve as a category-label-free reward for a separate explorer. Across our settings, learned scans reduce query MSE and can improve final-feature recognition over matched random walks. However, the rankings of scans differ for reconstruction and recognition, showing that lower query error does not by itself imply universally better recognition.

Direct query diagnostics cover one CCPL configuration, and cross-objective comparisons or controlled pretraining ablations remain unmeasured. Also, explorer training is privileged because full images provide unseen query targets, and development labels are used only for checkpoint selection, not for the reward. Finally, we do not systematically vary pretraining duration or explorer training budget, so gains from additional optimization remain unmeasured. Although scans are matched by glimpse budget, they differ in coverage and movement geometry; our acquisition comparisons therefore do not isolate scan ordering alone.

Because the predictor answers queries at arbitrary relative poses, CCPL is not tied to still images. Adding a temporal offset to the move would extend the interface to video, where the scene changes between glimpses, and a camera whose pan and zoom are physical actions would let the explorer act on a real sensor. The predictor is also a conditional generator of continuous latents: run along a chosen scan from a short history, it could generate images glimpse by glimpse, in any order and at any scale, in the spirit of continuous-token autoregressive generation (Li et al., 2024). Finally, letting the explorer choose every move, or rewarding the readout that is evaluated, would test how far learned acquisition can go beyond steering a random walk.

AI USE STATEMENT

Generative AI tools were used for research brainstorming, literature discovery, experiment and software design, code generation and review, interpreting results, figure planning, and drafting and editing parts of this manuscript. The authors verify AI-assisted claims against primary sources and experiment logs, validate generated code through review and testing, and take responsibility for the final text, claims, results, and artifacts.

REPRODUCIBILITY STATEMENT

Appendix A lists the code, data, training, and evaluation artifacts required to reproduce every reported result. Source code and configuration files will be released as open source.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024. doi: 10.18653/v1/2024.acl-long.662.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629, 2023. doi: 10.48550/arXiv.2301.08243.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. doi: 10.48550/arXiv.2506.09985.

Ruzena Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, 1988. doi: 10.1109/5.5968.

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025. doi: 10.48550/arXiv.2511.08544. URL https://arxiv.org/abs/2511.08544.

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022. doi: 10.48550/arXiv.2105.04906.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024. doi: 10.48550/arXiv.2404.08471.

Ricardo Buitrago and Albert Gu. Understanding and improving length generalization in recurrent models. In *International Conference on Machine Learning (ICML)*, volume 267, pp. 5856–5874. PMLR, 2025. URL https://proceedings.mlr.press/v267/buitrago25a.html.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. doi: 10.1109/ICCV48922.2021.00951.

Tianqi Chen, Ian Goodfellow, and Jonathon Shlens. Net2net: Accelerating learning via knowledge transfer. In *International Conference on Learning Representations (ICLR)*, 2016. doi: 10.48550/arXiv.1511.05641.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, 2020. doi: 10.48550/arXiv.2002.05709. URL https://proceedings.mlr.press/v119/chen20j.html.

Changho Choi, Minho Kim, and Jinkyu Kim. Mambaeye: A size-agnostic visual encoder with causal sequential processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 2659–2668, June 2026. doi: 10.48550/arXiv.2511.19963. URL https://arxiv.org/abs/2511.19963.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale.

In *International Conference on Learning Representations (ICLR)*, 2021. doi: 10.48550/arXiv.2010.11929.

Alaaeldin El-Nouby, Michal Klein, Shuangfei Zhai, Miguel Ángel Bautista, Vaishaal Shankar, Alexander T Toshev, Joshua M Susskind, and Armand Joulin. Scalable pre-training of large autoregressive image models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12371–12384. PMLR, 2024. doi: 10.48550/arXiv.2401.08541. URL https://proceedings.mlr.press/v235/el-nouby24a.html.

S. M. Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S. Morcos, Marta Garnelo, Avraham Ruderman, Andrei A. Rusu, Ivo Danihelka, Karol Gregor, David P. Reichert, Lars Buesing, Theophane Weber, Oriol Vinyals, Dan Rosenbaum, Neil Rabinowitz, Helen King, Chloe Hillier, Matt Botvinick, Daan Wierstra, Koray Kavukcuoglu, and Demis Hassabis. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018. doi: 10.1126/science.aar6170.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12873–12883, 2021. doi: 10.1109/CVPR46437.2021.01268.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, 2020. doi: 10.48550/arXiv.2006.07733.

Ali Hatamizadeh, Yejin Choi, and Jan Kautz. Gated deltanet-2: Decoupling erase and write in linear attention. *arXiv preprint arXiv:2605.22791*, 2026. doi: 10.48550/arXiv.2605.22791.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. doi: 10.1109/CVPR52688.2022.01553.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020. doi: 10.48550/arXiv.2006.11239.

Dinesh Jayaraman and Kristen Grauman. Learning to look around: Intelligently exploring unseen environments for unknown tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1238–1247, 2018. doi: 10.1109/CVPR.2018.00135.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*, 2018. doi: 10.48550/arXiv.1710.10196.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014. doi: 10.48550/arXiv.1312.6114.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In *Advances in Neural Information Processing Systems*, 2024. doi: 10.48550/arXiv.2406.11838.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. doi: 10.48550/arXiv.2210.02747.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. doi: 10.48550/arXiv.1711.05101.

Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. Recurrent models of visual attention. *arXiv preprint arXiv:1406.6247*, 2014. doi: 10.48550/arXiv.1406.6247.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729. IEEE, 2008. doi: 10.1109/icvgip.2008.47.

Adam Pardyl, Michał Wronka, Maciej Wołczyk, Kamil Adamczewski, Tomasz Trzciński, and Bartosz Zieliński. Adaglimpse: Active visual exploration with arbitrary glimpse position and scale. In *European Conference on Computer Vision (ECCV)*, pp. 112–129, 2024. doi: 10.1007/978-3-031-72664-4_7.

Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2536–2544, 2016. doi: 10.48550/arXiv.1604.07379.

Santhosh K. Ramakrishnan, Dinesh Jayaraman, and Kristen Grauman. Emergence of exploratory look-around behaviors through active observation completion. *Science Robotics*, 4(30), 2019. doi: 10.1126/scirobotics.aaw6326.

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

Soroush Seifi and Tinne Tuytelaars. Where to look next: Unsupervised active visual exploration on 360° input. *arXiv preprint arXiv:1909.10304*, 2019. doi: 10.48550/arXiv.1909.10304.

Soroush Seifi, Abhishek Jha, and Tinne Tuytelaars. Glimpse-attend-and-explore: Self-attention for active visual exploration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16137–16146, 2021. doi: 10.48550/arXiv.2108.11717.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. doi: 10.48550/arXiv.1807.03748.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992. doi: 10.1007/BF00992696.

A Reproducibility Checklist

- Code: the explorer, the full-ImageNet probe, the query-capability evaluation, and the efficiency measurement. The corresponding source will be released as open source.
- Dataset source, split, licenses, and preprocessing: ImageNet-1k ImageFolder layout. Oxford 102 Flowers is the transfer evaluation in Appendix G; every component stays frozen and only the linear heads are refit.
- SSL trajectories: a pretraining random walk with reflect-padded SigRand pan plus log-scale Brownian motion at Q=1.023 with physical reflection (Section 3.1). Each image is randomly cropped (area $[0.6, 1]$, aspect $[3/4, 4/3]$), horizontally flipped with probability 0.5, and resized with its aspect ratio kept so that its long side is drawn uniformly from 256 to 1024 pixels; each sequence has 1024 glimpses.
- Latent codec: frozen convolutional patch VAE, 128-D deterministic mean latents. Architecture, data, optimizer, loss, and the completed run are in Appendix C.
- Observers: Gated DeltaNet-2 256w6d, 512w12d and 1024w24d, each grown to $\phi_{16,1280}$. The acquisition comparisons use the final 512w12d and 1024w24d checkpoints.
- Predictors: jump2 $\phi_{4,256} \to \phi_{8,512} \to \phi_{12,1024} \to \phi_{16,1280}$ (Appendix F).
- Explorer: canvas 256, patch 16, start at the canvas centre with s=1; pan unclamped, samples past the image edge read from a mirrored halo, scale bounds and reflection inherited from the pretraining checkpoint. Budget T=1024 glimpses. Learned moves land at glimpses $8, 16, \ldots, 1024$ (128 decisions), $\tanh$-bounded to $\mathrm{max_pan}$=4 glimpse widths and $\mathrm{max_log_zoom}$=0.35; the other moves are unclipped increments of the pretraining random walk (log-uniform pan scale, log-scale Brownian motion over the full episode). Non-policy random-walk segments are observed in blocks of 4 (512w12d) or 64 (1024w24d) glimpses, checked against serial recurrence. Training crops are random resized crops (area $[0.6, 1]$, aspect $[3/4, 4/3]$) with horizontal flips. Terminal reward only: latent query MSE over 16 jittered-grid queries per scale in $\{0.5, 1, 2, 4\}$, the mean of two samples after 8 Euler steps from noise, never decoded, divided by the pretraining random walk's mean error on 256 centre-cropped training images, computed before training. Policy: two-layer Gated DeltaNet-2 trunk (width 256, four heads), $\tanh$-squashed Gaussian head with learned standard deviation (initially 0.5); input $\mathrm{LayerNorm}(h_t)$, displacement since the previous decision (translation in that decision's glimpse widths), remaining-budget fraction. REINFORCE (Williams, 1992) with a leave-one-out baseline (Ahmadian et al., 2024) over two rollouts per image that share crop, walk increments, queries and noise; advantages standardized over the global batch; loss averaged over the 128 learned actions; entropy coefficient 0.01; no critic. Adam, learning rate 10^{-4}, ϵ=10^{-5}, gradient-norm clip 1.0; one update per cycle of 64 images (128 rollouts), 1800 cycles, bfloat16. Checkpoints every 100 cycles.
- Optimization: SSL global batch 1024, bf16-mixed, AdamW peak learning rate 3×10^{-4} (Loshchilov & Hutter, 2019); β_1=0.9, β_2=0.999, ϵ=10^{-8}, weight decay 0.05 with normalization and bias parameters exempt; gradient-norm clip 1.0. Growth schedules and accumulation are in Appendix F.
- Explorer training time is reported in Appendix H.3. Evaluation latency and peak device memory are in Table 9.
- Seeds, checkpoint selection, and uncertainty: one SSL seed and one explorer seed per observer. Explorer checkpoints are probed every 300 cycles on a development split (5 training images per class, held out from explorer training; heads fit on 50 others per class); the one with the highest final-feature top-1 is evaluated—cycle 900 for 512w12d and 1200 for 1024w24d—and was fixed before validation was scored. Paired 95% image-bootstrap intervals use $10{,}000$ resamples (seed 17); they cover evaluation images, not training seeds.
- Nonlinear readout: on the cached 1024w24d features only, a two-layer head (linear, GELU, dropout, linear; hidden width 2048) and a linear head on the concatenated final and mean features. Hyperparameters are chosen on $51{,}247$ held-out training images of the random-walk arm and then shared across scans (Appendix H.2).

- Evaluation: acquisition heads are fit on all $1{,}281{,}167$ ImageNet-1k training images and scored on the $50{,}000$ validation images, each resized to short side 293 and centre-cropped to 256; final and mean features from one 1024-glimpse pass; per-vector LayerNorm, then training-set standardization; AdamW, learning rate 0.03, weight decay 10^{-4}, batch 8192, 100 epochs, step decay $\times 0.1$ at epochs 60 and 80, final epoch scored once. Learned scans sample actions under recorded seeds with a fixed extraction batch layout, since a closed-loop scan is sensitive to batch composition at rounding level. Reconstruction uses the same validation images and each scan's cached final feature and pose, with the reward's query design and paired noise. ω^2 on the $\sigma{=}4$ lineage; walk-last $1024/2048/4096$ not logged on jump2.

B EVIDENCE SCOPE

Table 4 separates completed measurements from open questions. Recognition results do not by themselves establish reconstruction gains, content-dependent acquisition, or policy transfer.

Analysis	Status	Evidence and boundary
Acquisition recognition	Measured	The learned scan, matched random walk, and all nine lattice scans at $T{=}1024$ (Table 3). Final-feature gains over matched random walks accompany mean-feature losses; stronger lattice scans remain. A two-layer readout on the same 1024w24d features leaves that pattern in place (Appendix H.2).
Checkpoint selection	Documented	Development-split selection and training cost (Appendix H.3); selection precedes official-validation scoring.
Matched reconstruction	Measured	Lower latent/pixel terminal query MSE than matched random walks on both CCPL models, with paired image intervals (Appendix H.4).
Direct query tests	Measured	1024w24d history prefixes, scale sweep, memory reset and two query-move shuffle controls (Table 2); fully unobserved footprints form a small subset.

Table 4: Scope of the available evidence. Missing measurements do not count as negative results; reconstruction and recognition are evaluated separately.

C FROZEN IMAGE PATCH CODEC

The three recorded pretraining configurations specify the same convolutional patch-VAE artifact. Its encoder takes an RGB 16×16 glimpse and uses three convolution/GELU stages with output channels $64, 128, 256$, kernel sizes $2, 2, 1$, and strides $2, 2, 1$, respectively, with no convolutional padding. The resulting $256 \times 4 \times 4$ feature is flattened and projected from 4096 to 512 channels, refined by two SelfHyper blocks of width and rank 512, and projected to 256 outputs. The frozen encoder uses the first 128 outputs, the posterior mean $\mu(x)$. SelfHyper is defined in Appendix D.1. Padding-free codec convolutions are separate from the reflective image boundary used to extract glimpses.

The decoder maps 128 latent channels to 512, applies two further SelfHyper blocks, and projects to 4096 channels. After reshaping to $256 \times 4 \times 4$, two transposed convolutions with kernel and stride 2 produce $128 \times 8 \times 8$ and $64 \times 16 \times 16$ features, each followed by GELU. A 1×1 convolution produces RGB values, followed by $(1 + \tanh(\cdot))/2$. The codec applies one global latent scale, rather than per-channel standardization, as in Equation 6:

$$E(x) = \mu(x)/c, \qquad D(z) = D_{\mathrm{VAE}}(cz), \qquad c = 0.7959760427474976. \tag{6}$$

CCPL uses these deterministic mean encodings, without posterior sampling. Encoder and decoder parameters are frozen throughout observer/predictor pretraining, explorer training, and evaluation.

The corresponding codec training run completed 60 epochs (numbered 0–59) and $4{,}500$ optimizer steps, with 75 steps per epoch on eight GPUs. Its configuration records a per-worker batch of $16{,}384$

(effective batch 131,072), peak learning rate 0.005, `beta`= 0.001, and `warmup_frac`= 0.2. The logged learning rate rose to 0.005 at epoch 11 and decayed to zero at epoch 59. The run logged reconstruction MSE and KL separately; at the last epoch these were 0.00194 and 0.945, respectively. The saved checkpoint was selected by the run's best logged held-out PSNR, 30.73 dB at epoch 30; the last-epoch held-out value was 30.71 dB. These are codec-run diagnostics, distinct from the spatial-query MSE measurements in Table 3.

Training patches are ROI-aligned 16×16 samples from ImageNet-1k train, and the held-out PSNR is computed on sampled patches from the ImageNet-1k validation set. The optimizer is AdamW. The objective is reconstruction MSE plus `beta` $\times$ KL, with `beta` $= 0.001$. The KL term is a mean over the 128 latent channels. This offline codec stage is separate from the 208-epoch CCPL schedule in Appendix F.

D Conditioning and query generation

The observer concatenates the glimpse latent and executed-move embedding $[z_t; e(a_t)]$, then applies a Linear/SiLU projection before its recurrent layers. The predictor instead combines $\mathrm{LN}(h_t)$, the planned-move embedding and flow time through SelfHyper conditioning, which is injected into a separate noisy-latent denoiser stream. The clean prediction target never enters the observer's memories.

Query generation holds the history feature fixed and integrates the frozen predictor from Gaussian noise, conditioned on the relative query move. The explorer reward compares predicted and target latents directly; the frozen codec decoder is used for pixel-space evaluation and visualization.

D.1 SelfHyper implementation

SelfHyper is an input-conditioned multiplicative refinement used in both the codec and the predictor. For a column vector $u \in \mathbb{R}^D$, its shared algebraic form is

$$Q(u) = U\big[g(u) \odot (V^\top u)\big] + b(u), \qquad U, V \in \mathbb{R}^{D\times r}. \tag{7}$$

In Equation 7, $g : \mathbb{R}^D \to \mathbb{R}^r$ and $b : \mathbb{R}^D \to \mathbb{R}^D$ are separate two-layer MLPs with GELU and hidden width r. The gate and projected input interact by elementwise multiplication. The nonlinear gate means this is not simply a quadratic polynomial. The codec and predictor use separate parameters and different residual and normalization wrappers around this operation.

Each codec block computes $\mathrm{LN}_{\mathrm{out}}[x + \gamma \odot Q(\mathrm{LN}_{\mathrm{in}}(x))]$, with $D = r = 512$. The channelwise gain γ is initialized to zero; the block still applies its output LayerNorm at initialization and therefore is not an exact identity map.

In each predictor residual block, the conditioning vector is $m_0 = [\mathrm{LN}(h_t); e(a_{t+1}); e_\tau(\tau)]$. The movement embedding has 384 channels and the time embedding has 128; the latter is a sinusoidal embedding followed by a two-layer Linear/SiLU/Linear network. Thus m_0 has 768, 1024, or 1536 channels for the three observer widths. No additional projection changes this conditioning width. There are three successive SelfHyper refinements per residual block. The first computes

$$m_1 = \rho(m_0)\,\mathrm{LN}_{\mathrm{out}}[m_0 + Q(\mathrm{LN}_{\mathrm{in}}(m_0))]\,, \qquad \rho(m) = \max\left(\sqrt{\frac{1}{D}\sum_i m_i^2}, 10^{-6}\right). \tag{8}$$

The next two refinements follow Equation 8, but multiply the Q term by a learned channelwise gain initialized to zero. Each restores the RMS of its own input after LayerNorm. Parameters are separate across refinements and across predictor blocks. The conditioning rank r is the second index in $\phi_{d,r}$; its growth from 256 to 1280 is specified in Appendix F.

The refined condition m_3 modulates a separate denoising stream of width 1024. After time-dependent shift and scale of its normalized activation v, the block forms $\delta = U_o[g_o(m_3) \odot (V_o^\top v)] + b_o(m_3)$. Here the operator rank is 512, and both condition-to-operator MLPs have hidden width 256; these are distinct from the conditioning rank r. RMSNorm and a learned channelwise gain initialized to one normalize δ, which is added to v through a sigmoid time gate. A GELU feed-forward network with hidden width 2048 then supplies a residual update with a zero-initialized time-dependent output gate.

Each block also applies a SelfHyper update directly to the denoising stream: $x \leftarrow x + \gamma_x \odot \mathrm{RMSNorm}[Q_x(\mathrm{LN}(x))]$, using width 1024, rank and MLP hidden width 256, and zero-initialized γ_x. This residual sublayer is an exact identity at initialization. Multiplicative SelfHyper and operator products are evaluated in float32 under mixed-precision training. Final normalization, time modulation, and a linear projection produce the 128-channel velocity prediction.

E SSL GEOMETRY AND MEMORY DETAILS

E.1 SSL SCALE-WALK GEOMETRY

Pretraining trajectories follow one random-walk family. Planar pan uses per-step log-uniform spatial scale (sigrand), $\sigma \sim \mathrm{LogUniform}(0.5, 4)$ relative to the base spatial factor. Zoom is a log-scale Brownian motion at *fixed episode quadratic variation*, so sequence length is a discretization of a fixed physical zoom experiment rather than extra diffusion time. Let $q = \log s$. Combined pan-and-zoom steps give every token the scale increment of Equation 9,

$$\Delta q_k \sim \mathcal{N}\big(0,\, Q/(T-1)\big). \tag{9}$$

We pin Q=1.023, matching the historical Bernoulli-zoom quadratic variation $p_{\mathrm{zoom}}(T-1)\sigma^2$ at p=0.1, σ=0.1, and T=1024. That number is a duration pin, not a preferred occupancy band. Walls are the sensor interval after crop and aspect-fit, not a hand-set interval such as $[0.5, 4]$. Writing content and src for the short sides of the fitted rectangle and the native image,

$$s_{\min} = \max\big(1/P,\ \mathrm{content}/\mathrm{src}\big), \qquad s_{\max} = \mathrm{content}/P. \tag{10}$$

The fine wall is one native source pixel per fovea pixel (with $1/P$ only as box degeneracy); the coarse wall is one fitted image in the fovea. Overshoot in q is reflected, not clipped onto the endpoint. A start band such as $[0.5, 4]$ remains a legitimate initial condition; the walk is then folded into Equation 10. An embedding $e(a_t)$ expands the raw three-vector into a sinusoidal feature used by both the encoder and the predictor.

E.2 FLOW SOLVER AND NOISE SCHEDULE

On every pretrain the interpolation time is drawn from a logit-normal with location 0.6 and unit scale, $\tau = \sigma(0.6 + \varepsilon)$ with $\varepsilon \sim \mathcal{N}(0, 1)$, placing more mass on high-noise interpolants than a uniform draw on $(0, 1)$. Query generation integrates the learned velocity with 8 Euler steps from τ=1 to τ=0, and averages two latent samples before the latent error of Equation 4 is taken. Pixel evaluation instead decodes each sample, clips it to $[0, 1]$, and averages the decoded samples before comparison with the extracted query pixels.

E.3 RECURRENT MEMORY UPDATE

The Gated DeltaNet-2 update at layer ℓ *decouples* erase from write:

$$S_t^{(\ell)} = \big(I - k_t(b_t \odot k_t)^\top\big)\,\mathrm{Diag}\big(\exp(g_t)\big)\,S_{t-1}^{(\ell)} + k_t(w_t \odot v_t)^\top. \tag{11}$$

Here b_t is a channel-wise erase gate on the key axis, w_t is a channel-wise write gate on the value axis, and g_t is a channel-wise log-decay. Inference memory is constant in sequence length. The layer input is the concatenation of the glimpse latent and the current move embedding, so erase and write can, in principle, treat path coordinates differently from scene content. That is a memory-edit hypothesis, not a claim that Equation 11 yields class-aligned features.

F PREDICTOR MORPH RECIPE

Figure 4 shows the function-preserving expansion used in each predictor-growth hop. Residual MLP width, SelfHyper refinement depth, and the remaining conditioner keys stay fixed. Only $\phi_{4,256}$ trains from scratch (100 epochs: 20-epoch warmup to peak learning rate 3×10^{-4}, then the original 300-epoch cosine truncated at epoch 100). Each hop: copy, identity-append, zero-pad, identity check, **1** new-only epoch, **5** predictor-only adapt epochs, then **30** joint epochs with a hop-local schedule.

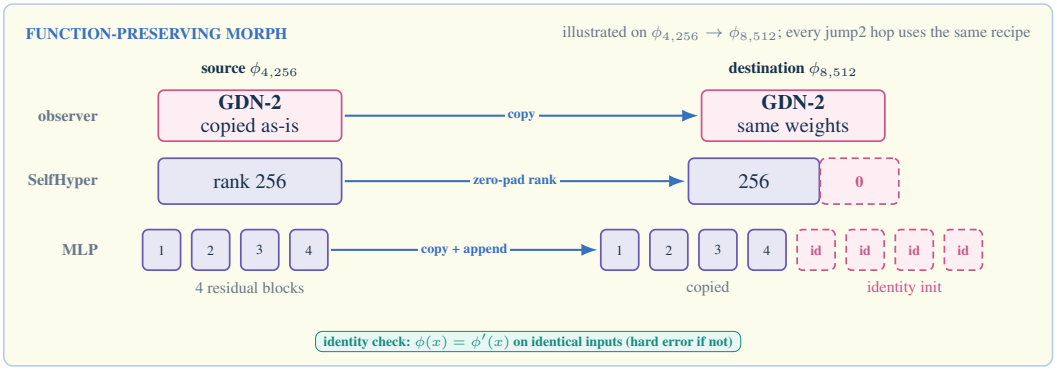

Figure 4: **Function-preserving predictor expansion.** The morph from $\phi_{4,256}$ to $\phi_{8,512}$ copies the observer and existing residual blocks, appends identity-initialized residuals, and zero-pads SelfHyper rank. Source and destination outputs must match on identical inputs before any optimizer step. The same construction applies at each growth step.

Conceptual budget $100 + 3 \times 36 = 208$ epochs per observer. Global batch stays 1024 over eight GPUs. For 256w6d and 512w12d, microbatch/accumulation is $128/1$ at the seed, then $64/2$, $32/4$, and $16/8$. For 1024w24d it is $32/4$, $32/4$, $16/8$, and $16/8$, respectively.

Denoiser parameter counts (millions; counted 2026-09-04) depend on observer width because the denoiser conditions on h_t:

			Denoiser params	
Predictor	MLP depth d	SelfHyper rank r	256w6d	512w12d
$\phi_{4,256}$	4	256	45.0M	49.5M
$\phi_{8,512}$	8	512	117.8M	134.6M
$\phi_{12,1024}$	12	1024	275.5M	324.3M
$\phi_{16,1280}$	16	1280	442.7M	523.5M

F.1 Pretraining loss histories

Figure 5 documents optimization across the complete progressive pretraining schedules for all three observers. Each curve uses the recorded loss history, including the predictor-only adaptation phases; no additional smoothing is applied. Growth changes predictor capacity and restarts the optimization schedule.

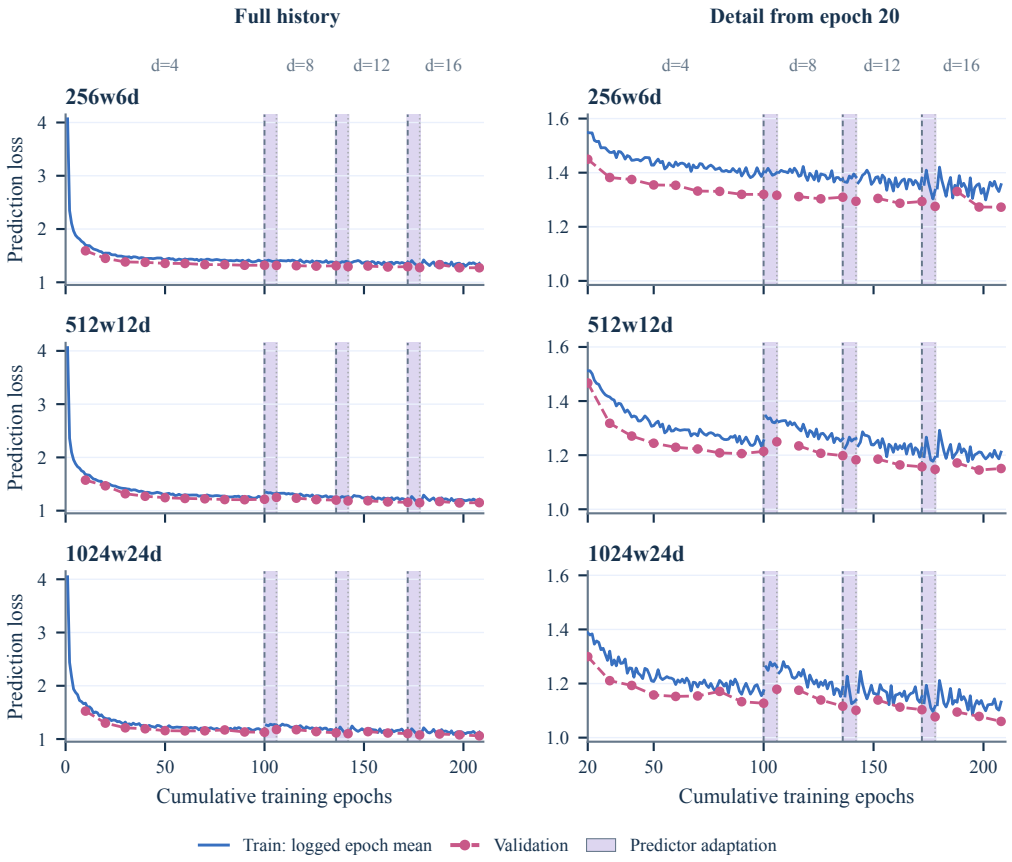

Figure 5: **Pretraining prediction loss.** Rows show the three observer configurations. Left: complete 208-epoch schedules; right: detail from epoch 20. Training curves average logged loss samples within each epoch, rather than all training batches; validation markers show recorded epoch losses. Dashed vertical lines mark predictor growth at epochs $100, 136, 172$; shaded bands mark the six-epoch adaptation phases, and dotted boundaries start the joint-training schedules. d denotes predictor depth. Lines are not joined across runs. These are pretraining flow-matching losses, distinct from terminal spatial-query MSE.

G TRANSFER TO OXFORD 102 FLOWERS

This section asks whether the acquisition effect measured on ImageNet carries over to a dataset none of the components were trained on. The codec, observer, predictor and explorer stay frozen at the ImageNet checkpoints selected in Section 4.1 (cycle 900 for 512w12d, 1200 for 1024w24d); only linear heads are fit on the new dataset.

Data and splits. We use all 8,189 images of Oxford 102 Flowers (Nilsback & Zisserman, 2008). The main result uses the standard split: heads fit on the union of the official training and validation sets (2,040 images: 10 training and 10 validation images per class) and are scored on the official test set (6,149 images). We also report a second, roughly 8:1:1 re-split that fits 7,370 images and tests 819 (Table 6). Its training images include 4,911 of the standard test images, so its accuracies are not comparable to published Flowers-102 results, and its small test set gives wide intervals.

Protocol. Extraction follows the ImageNet recognition protocol: one deterministic 256×256 centre crop per image, 1024 glimpses from the centre at $s=1$, bfloat16 with blocks of 4 (512w12d) or 64 (1024w24d), and the same action and query seeds. We compare the learned explorer, the matched random walk, and the two lattice scans with the highest ImageNet final-feature accuracy, Golden and Spiral. Heads use the ImageNet recipe unchanged (Section 3.5). No hyperparameter is selected, so validation images join the fit set. With at most 7,370 fit images and batch 8192, each epoch is a single full-batch step. Differences from the matched random walk are paired on the same test images, with 95% image-bootstrap intervals from 10,000 resamples.

Results. On the standard split, the direction of the ImageNet acquisition effect transfers (Table 5). Learned scanning raises final-feature top-1 over the matched random walk by 3.24 points on 1024w24d ($[1.85, 4.55]$). On 512w12d the gain is 1.30 points, and its interval $[-0.15, 2.72]$ includes zero. As on ImageNet, the mean-feature readout moves the other way: learned scanning is 6.07 and 2.75 points below the matched random walk. The second split shows the same signs, with larger final-feature gains of 5.01 and 7.94 points (both intervals exclude zero).

Lattice scans dominate much more strongly here than on ImageNet. On the standard split, Golden scanning exceeds the matched random walk by 26.00 and 25.13 final-feature points and by 3.87 and 5.85 mean-feature points. It exceeds the learned explorer on both readouts and both CCPL models. Spiral scanning also gains 20.85 and 15.19 final-feature points over the matched random walk, but falls below it on the mean feature. The small fit set may amplify the final-feature overfitting of the lattice-scan linear recipe observed on ImageNet, and the explorer was trained only on ImageNet. These results therefore support transfer of the learned-versus-random-walk ordering on the final feature.

	512w12d		1024w24d	
Scan	Final	Mean	Final	Mean
Random walk	31.05	63.90	25.66	62.27
Learned	32.35	57.83	28.90	59.52
Golden	57.05	67.77	50.79	68.12
Spiral	51.89	59.59	40.85	59.99

Table 5: Oxford 102 Flowers linear-probe top-1 (%) on the standard split (fit 2,040, test 6,149) at $T=1024$ glimpses. All components are frozen at their ImageNet checkpoints.

H ADDITIONAL EXPERIMENTAL RESULTS

H.1 LATTICE SCANNING PATTERNS

Table 3 reports all nine lattice patterns alongside the learned scan and matched random-walk references. A lattice path observes the centre first, then the first 1023 positions in a 32×32 lattice with 8-pixel stride, keeping each image's boundary-applied initial scale. The ordering determines which

	512w12d		1024w24d	
Scan	Final	Mean	Final	Mean
Random walk	47.37	74.36	41.39	77.17
Learned	52.38	72.89	49.33	75.09
Golden	67.52	79.00	66.91	80.71
Spiral	65.69	72.41	61.78	77.17

Table 6: Oxford 102 Flowers top-1 (%) on the 8:1:1 re-split (fit 7,370, test 819). Its fit set contains 4,911 standard test images, so it is not comparable to published results.

final lattice cell is omitted; these paths do not have exactly identical coverage. The Shuffled grid samples a lattice order per image under a recorded seed. Raster and snake variants traverse the same lattice in different orders. Golden (Choi et al., 2026) uses a golden-ratio stride through the flattened lattice, adjusted to be coprime to its size so that it visits every lattice cell before repeating. All lattice patterns hold scale constant.

All heads use the full-data protocol in Appendix A. The 512w12d lattice scans use float32 with observation blocks of 64, so their same-regime random-walk reference reaches 16.54% final-feature and 26.72% mean-feature top-1 in that regime, with latent MSE 0.873 and pixel MSE 0.0352. Its similar recognition scores to the bfloat16/block-4 matched random walk do not establish numerical invariance of the closed-loop learned policy. All 1024w24d scans use bfloat16/block-64.

H.2 Two-layer readout on frozen features

This section asks whether the 1024w24d acquisition pattern in Table 3 is specific to a linear head. The codec, observer, predictor, and explorer stay frozen at the cycle-1200 checkpoint. Heads read the same cached final and mean features as the linear probes, from one 1024-glimpse pass on each of the 50,000 validation images. The 512w12d observer is not included.

The two-layer head is a linear layer, a GELU, dropout, and a linear layer, with hidden width 2048. It is trained for 100 epochs at batch size 8192, learning rate 3×10^{-4}, dropout 0.7, and weight decay 0.2. A separate linear head reads the concatenation of the final and mean features, with learning rate 3×10^{-3} and weight decay 0.5. Both settings were selected by top-1 accuracy on 51,247 held-out training images from the random-walk arm (every training image whose index is divisible by 25), then used unchanged for every scan. Official validation images were not used for that choice.

Table 3 reports the two-layer and Concat heads for every scan. Against the matched random walk, the learned scan's two-layer final-feature advantage is 1.56 points (paired 95% image-bootstrap interval $[+1.21, +1.92]$), its two-layer mean-feature change is -0.52 points ($[-0.83, -0.20]$), and its Concat change is -0.14 points ($[-0.48, +0.21]$).

H.3 Explorer checkpoint selection and training cost

Every 300 explorer updates, development heads fit 50,000 training images and score 5,000 held-out training images. These use a lightweight recipe separate from the full-data evaluation in Table 3. The highest development final-feature top-1 selected cycle 900 for 512w12d and cycle 1200 for 1024w24d; both selections were fixed before official validation was scored. Category labels therefore enter checkpoint selection, although they are absent from the explorer reward.

Both explorer runs completed 1800 updates and 230,400 trajectories. On four training GPUs per run, 512w12d required 9.11 wall-clock hours (36.46 GPU-hours), and 1024w24d required 10.19 hours (40.76 GPU-hours). These totals exclude the independent evaluators, full-data extraction, linear-head fitting and reconstruction decoding; they are not end-to-end training cost or deployment latency.

H.4 Matched reconstruction and paired uncertainty

Reconstruction uses the same 50,000 validation images as recognition, with cached raw terminal features and each scan's actual final pose. Absolute queries at scales $\{0.5, 1, 2, 4\}$, 16 per scale, and

generation noise are paired by image; the report verifies matching checksums across arms. Two predictions use 8 Euler steps each. Latent samples are averaged before squared error; for pixel MSE each sample is decoded and clipped to $[0, 1]$ before averaging. These are queried-patch errors, not dense image MSE.

Table 7 gives paired 95% image-bootstrap intervals (10,000 resamples, seed 17) for recognition. Learned scans improve final-feature recognition on both CCPL models relative to their matched random walks, and also lower terminal latent and pixel query MSE on both (Table 3); all four paired intervals of the MSE difference exclude zero. The 512w12d mean-feature loss excludes zero; the 1024w24d mean-feature interval includes zero. These intervals do not measure training-seed variability. Reconstruction comparisons use the newly evaluated random-walk arm, not the training reward's calibration denominator.

Observer	Outcome	Effect	95% interval
512w12d	Final top-1 (pp)	$+1.18$	$[+0.84, +1.52]$
	Mean top-1 (pp)	-2.70	$[-3.05, -2.36]$
1024w24d	Final top-1 (pp)	$+1.34$	$[+0.98, +1.71]$
	Mean top-1 (pp)	-0.36	$[-0.71, +0.01]$

Table 7: Learned versus matched random-walk acquisition on 50,000 validation images. Effects are learned minus random walk in top-1 percentage points; positive favors learned. Intervals are paired image bootstraps.

The reconstruction effect depends on query scale (Table 8). On 512w12d, learned scanning has higher finest-scale latent error than the random walk (0.616 against 0.604), despite lower overall error. Its pixel error is lower at every reported scale on both observers. The scale-specific results concern this terminal-query protocol, not a controlled test of unseen-scale generalization.

Observer	Scan	Space	$s = 0.5$	$s = 1$	$s = 2$	$s = 4$
512w12d	Random walk	Latent	0.604	0.812	0.974	1.098
	Random walk	Pixel	0.0385	0.0363	0.0339	0.0324
	Learned	Latent	0.616	0.804	0.941	1.020
	Learned	Pixel	0.0346	0.0312	0.0277	0.0241
	Random walk (float32)	Latent	0.604	0.813	0.976	1.099
	Random walk (float32)	Pixel	0.0385	0.0362	0.0339	0.0323
	Golden (float32)	Latent	0.661	0.891	1.103	1.322
	Golden (float32)	Pixel	0.0571	0.0571	0.0560	0.0555
1024w24d	Random walk	Latent	0.555	0.757	0.894	1.011
	Random walk	Pixel	0.0289	0.0270	0.0254	0.0282
	Learned	Latent	0.552	0.750	0.876	0.931
	Learned	Pixel	0.0258	0.0239	0.0215	0.0200
	Golden	Latent	0.665	0.902	1.124	1.380
	Golden	Pixel	0.0613	0.0611	0.0602	0.0617

Table 8: Per-scale terminal query MSE on 50,000 validation images, with latent MSE to three decimals and pixel MSE to four. On 512w12d, the random walk and learned scan use bfloat16/block-4, and Golden and its random-walk reference use float32/block-64; all 1024w24d scans use bfloat16/block-64. Each scan's paired interval against its random-walk reference excludes zero at every scale.

Golden has higher reconstruction error than its same-regime random-walk reference at every reported query scale (Table 8), despite its strong final-feature recognition. Its aggregate latent and pixel MSE are 0.994 and 0.0564 against 0.873 and 0.0352 for its random-walk reference on 512w12d, and 1.018 and 0.0611 against 0.804 and 0.0274 on 1024w24d; all paired intervals, aggregate and per scale, exclude zero. The small observer's Golden/random-walk pair uses float32/block-64, distinct from the learned/random-walk bfloat16/block-4 pair. These results reinforce that reconstruction error and linear-probe accuracy rank scanning patterns differently.

H.5 EFFICIENCY MEASUREMENTS

Table 9 separates observation cost from optional query generation. Both stages are measured on one NVIDIA H100 80GB (PyTorch 2.11, CUDA 12.8, flash-linear-attention 0.5.2) in bf16 at batch size 16, with the same acquisition block size (64) for every row. Each measurement runs five warmup batches and then 50 timed batches bracketed by device synchronization; we report the median time per image. Observation time starts from preprocessed images on the device and includes crop extraction, the codec, observer, policy when used, and the classification head; query time starts from the cached terminal feature and includes the predictor and decoder. Peak accelerator memory is measured separately for the two stages, each in its own process with only that stage's weights resident. Queries are fixed at 64 per image (16 at each of the scales 0.5, 1, 2 and 4), with two generation samples and eight Euler steps. The learned scan is slower than the matched random walk because the policy decides every eight glimpses, so only the prescribed moves between decisions are processed as a block. Neither constant-size recurrent memory nor a matched observation budget alone establishes a measured speed or memory advantage.

		Observation		Queries	
Backbone	Input / scan	ms/image	GiB	ms/image	GiB
512w12d	Random walk	26.6	0.30	152.2	2.05
512w12d	Learned	248.0	0.28	145.6	2.05
1024w24d	Random walk	40.3	1.23	146.4	2.67
1024w24d	Learned	455.0	1.21	143.9	2.67

Table 9: **Inference cost.** One H100, bf16, batch size 16, block size 64. All rows use 1024 glimpses. Observation and query columns report median milliseconds per image and peak accelerator memory in GiB, including resident weights.

H.6 SPATIAL-QUERY PROTOCOL AND CONDITIONING DIAGNOSTICS

Table 2 uses the same frozen 1024w24d CCPL model as the acquisition comparison, with the pretraining random walk alone. An explorer checkpoint supplies configuration only; no learned action is used. Each validation image is resized to short side 293, centre-cropped to 256, and observed at history lengths $64, 128, 256, 512, 1024$ along one recorded trajectory. Evaluation uses bfloat16, observation blocks of 64, and batches of 32 per process across eight processes. The action seed is 2000020, and the fresh query/noise seed is 7000020.

Each image has 16 jittered-grid queries at each of four scales $\{0.5, 1, 2, 4\}$, held fixed across prefixes and controls. Each query averages two samples generated from Gaussian noise with eight Euler steps. Latents are averaged before latent MSE; decoded samples are individually clamped to $[0, 1]$, averaged, then compared with the raw queried pixels. All-query errors average dimensions, queries and images. Unobserved-query errors instead pool eligible queries across images. Image-bootstrap intervals use $10{,}000$ resamples with seed 17, taking the 2.5th and 97.5th percentiles; paired controls resample the same images, and unobserved-query intervals resample image-level error sums and eligible counts together.

The final-glimpse control initializes fresh recurrent memories, processes the recorded final patch and its executed move, and queries from the recorded final pose. Both query-move shuffle controls keep the final feature, targets and noise fixed while applying a per-image derangement to query-move conditioning. One shuffles all 64 queries; the other shuffles within each 16-query scale group. Table 10 reports the paired increases in error. The within-scale control changes spatial conditioning without swapping scales; all six reported intervals exclude zero.

Eligibility is defined by source-pixel support boxes of side $16s$, folded through reflection. A query is unobserved only if its box shares no source pixel with the union of observed glimpse boxes. The common subset uses the full 1024-glimpse mask for every prefix and control, including the memory reset. It contains 3748 queries across 2598 images, or 0.12% of all queries. At scales $0.5, 1, 2, 4$, the respective query/image counts are $2614/2158$, $885/760$, $202/165$ and $47/36$; image sets can overlap across scales. Consequently, the last scale is only a sparse diagnostic of wholly unobserved support.

	Latent MSE difference		Pixel MSE difference	
Control	Mean	95% CI	Mean	95% CI
Final-glimpse reset	0.313	[0.311, 0.314]	0.0557	[0.0553, 0.0560]
Query-move shuffle (all)	0.398	[0.396, 0.400]	0.0639	[0.0635, 0.0643]
Query-move shuffle (within scale)	0.398	[0.396, 0.400]	0.0657	[0.0653, 0.0662]

Table 10: Conditioning perturbations minus the full 1024-glimpse history: paired mean MSE differences and 95% image-bootstrap intervals on 50,000 validation images.

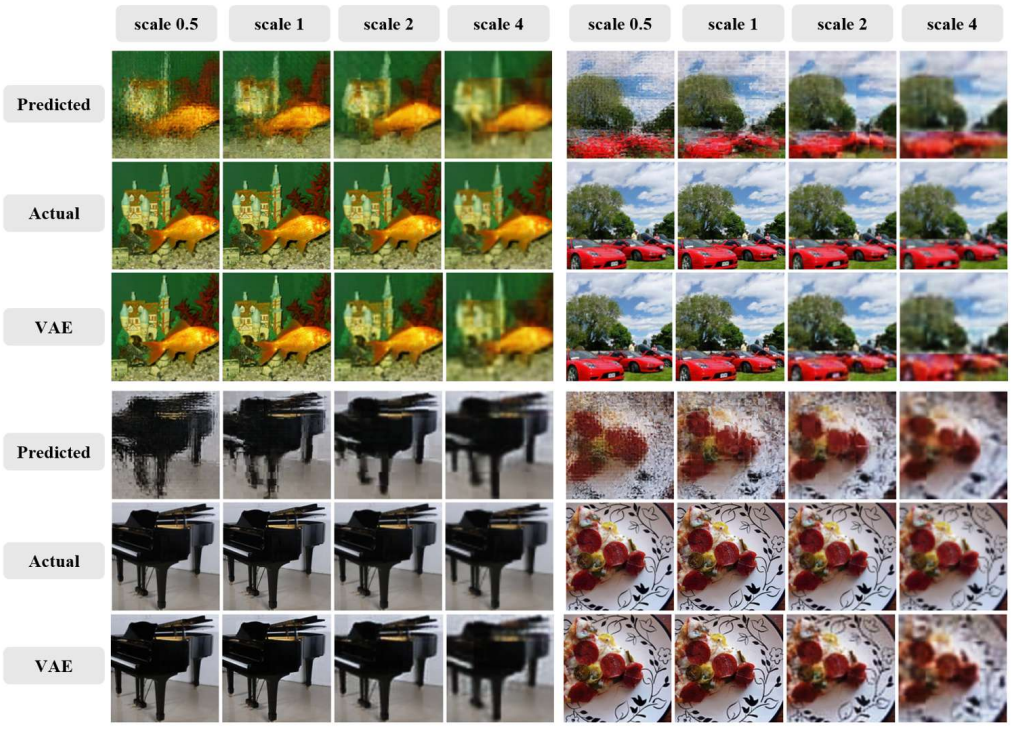

Figure 6: **Multiscale prediction from a single latent vector.** Predicted mosaics from the final feature of the 1024w24d explorer after 1024 glimpses, and the ground-truth image (labeled "actual") at query scales $0.5, 1, 2, 4$. The same latent vector conditions all predicted panels. The row labeled "VAE" is the frozen codec's reconstruction of the ground truth, so blur shared with that row comes from the codec rather than the prediction.

Mean canvas coverage rises from 21.43% at 64 glimpses to 98.33% at 1024. If eligibility is recomputed at each history length, counts shrink from 1,544,144 through 874,350, 312,718 and 55,054 to 3748. Errors on those changing subsets are not a matched comparison across history lengths; Table 2 therefore uses only the common subset.

I Supplementary multiscale prediction

Figure 6 uses the final feature of the 1024w24d explorer after 1024 glimpses; the row labeled "actual" is the ground-truth image. The scan-reconstruction comparison appears in the main text as Figure 3. Neither figure replaces the aggregate measurements in Tables 2 and 3.

J ADDITIONAL RELATED-WORK CONTEXT

J.1 PREDICTIVE SELF-SUPERVISED LEARNING

Modern visual SSL organizes around a few supervision geometries. Contrastive and distillation methods such as SimCLR, BYOL and DINO learn invariance across augmented views of a fixed image (Chen et al., 2020; Grill et al., 2020; Caron et al., 2021). Masked autoencoders reconstruct missing pixels from a sparse visible context (He et al., 2022). Joint-embedding predictive architectures replace pixel reconstruction with prediction in representation space: I-JEPA predicts masked target embeddings from a context encoder (Assran et al., 2023), while V-JEPA extends feature prediction to video (Bardes et al., 2024). Non-contrastive regularizers such as VICReg stabilize joint-embedding training without negatives (Bardes et al., 2022). LeJEPA combines a JEPA predictive loss with Sketched Isotropic Gaussian Regularization (Balestriero & LeCun, 2025).

CCPL shares the predictive-learning motivation of these methods, but not the joint-embedding implementation: it models frozen codec latents with a generative flow objective, where I-JEPA predicts learned target embeddings and describes itself as non-generative. Its context is also a causal glimpse trajectory rather than a masked crop, a multi-view embedding match, or a passively observed video clip, and its conditioning variable is a continuous planned movement. Table 1 places these side by side. The relevant comparison is therefore not only the predictive loss, but whether a frozen predictor can score unobserved locations well enough to train acquisition.

Two rows deserve emphasis. CCPL's pretraining order follows a random walk, not a learned policy, so on that axis it is no more adaptive than a random mask; what differs is that its conditioning variable is a continuous movement, which is what later permits querying at poses no trajectory supplied. And GQN already conditions generation on a query viewpoint, so queryability is not new either—what CCPL adds is accumulating that queryable state causally, one partial glimpse at a time, and then reusing it to choose the next observation.

J.2 LATENT WORLD MODELS AND RECURRENT MIXERS

Latent dynamics models predict compressed future observations. Denoising diffusion models and flow matching provide conditional generative objectives (Ho et al., 2020; Lipman et al., 2023). Our realization uses conditional flow matching over next-glimpse codec latents; flow matching itself is not the contribution.

Linear attention compresses history into a fixed-size recurrent state. Gated DeltaNet-2 stores a matrix state and updates it with a gated delta rule whose erase and write gates are channel-wise and independent (Hatamizadeh et al., 2026). Length failures in recurrent models have been linked to unexplored states; state passing broadens attainable recurrent states during training (Buitrago & Gu, 2025). In this paper, Gated DeltaNet-2 is the mixer inside the MambaEye encoder. Length generalization is a secondary diagnostic, not the primary evaluation.