# BECKMANN WORLD MODELS: DIRECT TERMINAL-MAP LEARNING FOR ONE-STEP ACTION-CONDITIONED PREDICTION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Action-conditioned world models must predict the consequences of proposed controls quickly enough to support repeated queries. We introduce Beckmann World Models (BWM), which learn a direct map from noise, visual history, and actions to future RGB frames without a pretrained generative teacher. Beckmann transport supplies a conservation-based training rule for this terminal map using sampled noise–future pairs, without a population of generated negatives. However, learning the map on interpolated inputs does not ensure accurate prediction from the pure noise used at deployment. We rectify this source-endpoint gap by proposing endpoint-aligned training: direct supervision of the source query, motion and temporal losses, and spatial feature supervision, supported by a generated-history regularizer. These additions train the same RGB map and preserve one network call per future chunk. On PushT and Robomimic Can, BWM reduces 64-frame MSE by $45$–$49\%$ relative to same-backbone regression at comparable inference cost, with measured latency of $21$–$24$ ms per native chunk. It also improves prediction over locally trained DriftWorld and policy-performance estimation, while the planning comparison shows no detected improvement. Cumulative training costs are disclosed because these are comparisons of complete recipes. The results identify a practical route from autonomous transport to accurate one-call world prediction. Anonymous project page: https://beckmann-world.pages.dev/.

## 1 INTRODUCTION

A visual world model predicts what a robot will observe after executing a proposed action sequence. Such predictions support visual planning, policy evaluation, and learning from imagined interaction (Finn et al., 2016; Ebert et al., 2018; Ha & Schmidhuber, 2018; Hafner et al., 2020; 2023). Recent video simulators extend these capabilities to richer environments and manipulation tasks (Yang et al., 2024; Bruce et al., 2024; Zhu et al., 2025b;a; Guo et al., 2026). Their practical value depends on both accuracy and query cost: evaluating many candidate actions requires many predicted futures, and a plausible video is insufficient if it places the manipulated object in the wrong configuration.

Diffusion and flow models offer expressive conditional generators, but standard samplers repeatedly evaluate a neural field (Ho et al., 2020; Song et al., 2021b; Lipman et al., 2023; Ho et al., 2022; Voleti et al., 2022). Distillation amortizes a pretrained generator into fewer evaluations (Salimans & Ho, 2022; Yin et al., 2024b;a). Consistency Models and MeanFlow also support direct training of one-step generators (Song et al., 2023; Song & Dhariwal, 2024; Geng et al., 2025; 2026). DriftWorld brings one-call generation to action-conditioned video through the Drifting objective (Deng et al., 2026; Lu et al., 2026a). Thus, one-step inference is already possible. The remaining question is which learned object and training signal make that single call accurately predict action consequences.

We investigate the *terminal map* of Beckmann Transport Models (BTM) (Lee et al., 2026), illustrated in Figure 1. For fixed visual history and actions, this map sends each source tensor directly to the endpoint of an autonomous transport. Points along the same transport trajectory have the same terminal prediction and the resulting stopped-Jacobian update can be estimated from individual noise–future pairs. Unlike DriftWorld's conditional attraction–repulsion field, it requires neither kernel comparisons nor a generated negative population for each condition. Unlike diffusion distillation, it



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.

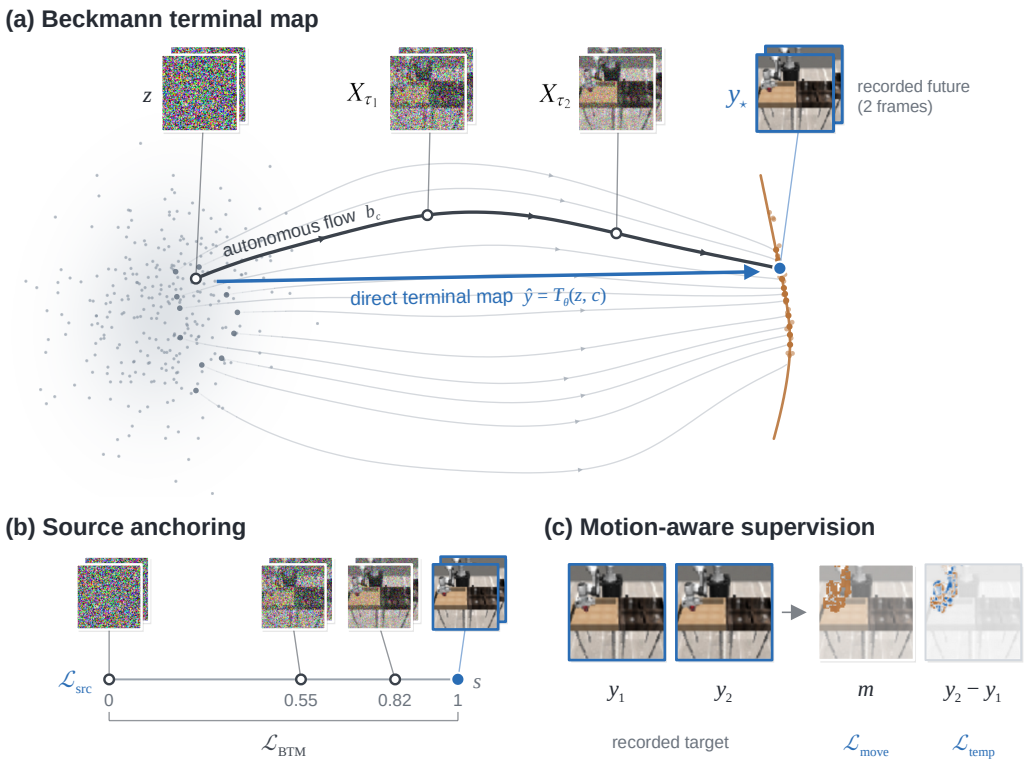



Figure 1: **Beckmann terminal maps and endpoint-aligned training.** (a) For a fixed condition $c = (h, a)$, an autonomous field $b_c$ (gray) carries noise samples to the data support. The terminal map is constant along each trajectory, so $z$, $X_{\tau_1}$, and $X_{\tau_2}$ share the endpoint $y_\star$; BWM evaluates the direct map $\hat{y} = T_\theta(z, c)$ (blue) once per chunk instead of integrating the flow. (b) $\mathcal{L}_{\text{BTM}}$ trains on the chord $x_s = (1 - s)z + sy$ and the boundary $s = 1$, but deployment queries only $s = 0$, which source anchoring supervises with $\mathcal{L}_{\text{src}}$. (c) $\mathcal{L}_{\text{move}}$ upweights error on moving pixels $m$ (from the target and the last observation), and $\mathcal{L}_{\text{temp}}$ matches the within-chunk difference $y_2 - y_1$. Thumbnails are two-frame Robomimic Can chunks; (a) is a 2D schematic whose $X_\tau$ thumbnails are the interpolants $x_{0.55}$ and $x_{0.82}$ from (b).

requires no pretrained generative teacher. The JVP and endpoint evaluations still have a measurable training cost.

The map directly predicts the future chunk needed by a planner, but translating BTM to this setting is insufficient. Its transport update samples mixed noise–data inputs, and its boundary loss acts on recorded futures; deployment queries pure noise. Our real-history probes reveal a substantial error gap between these input locations. Good reconstruction at the data boundary therefore cannot be used as evidence that the deployed source query is accurate.

We address this gap with *endpoint-aligned Beckmann training*. We retain the conservation-based update and explicitly supervise the pure-noise prediction. Motion and temporal losses emphasize changing robot and object pixels, while frozen spatial features supervise structure in the predicted frames. A detached generated-history regularizer supports autoregressive use. All components train the same map and leave inference unchanged. The design therefore improves what the one-call predictor learns without adding a test-time solver, a refinement network, or a generative teacher.

On PushT and Robomimic Can, the resulting BWM improves the measured prediction–latency tradeoff over same-split DriftWorld, MSE regression, and locally trained diffusion controls. It nearly halves 64-frame MSE relative to regression at comparable one-call latency. Endpoint probes and action swaps connect the gain to more accurate source predictions without losing action dependence. Policy-performance estimation also improves, although native GPC-RANK does not show a significant improvement. We disclose inherited training costs and distinguish released checkpoints.

Our contributions are:

- **A conditional Beckmann terminal map.** We formulate pair-based stopped-Jacobian training for RGB prediction and diagnose the gap between boundary and source accuracy.
- **Endpoint-aligned world-model training.** Source anchoring and motion, temporal, and spatial losses supervise the deployed map while retaining one-call inference.
- **Prediction, mechanism, and utility evidence.** Matched-backbone comparisons show $45$–$49\%$ lower MSE than regression, while source/action probes and visual recipe ablations connect the gains to endpoint accuracy and motion structure.

## 2 Related Work

**Diffusion and transport-based generation.** Diffusion models learn denoising dynamics, while Flow Matching and stochastic interpolants learn velocity fields transporting a source distribution to data (Ho et al., 2020; Song et al., 2021b; Lipman et al., 2023; Albergo et al., 2023). DDIM, EDM, and Rectified Flow improve sampling or path geometry (Song et al., 2021a; Karras et al., 2022; Liu et al., 2023). Latent representations and transformer backbones change computational cost and capacity (Rombach et al., 2022; Peebles & Xie, 2023). Our focus is the learned terminal prediction itself, rather than a numerical solver or a larger backbone.

**Distillation into direct generators.** Progressive Distillation halves a teacher's sampling steps; TRACT and Latent Consistency Models also learn large transitions from pretrained models (Salimans & Ho, 2022; Berthelot et al., 2023; Luo et al., 2023). DMD trains a student through real and generated score estimates, with paired regression for stability; DMD2 removes that regression requirement and improves distribution matching (Yin et al., 2024b;a). Adversarial Diffusion Distillation provides another teacher-based route (Sauer et al., 2024). BWM learns from recorded task transitions without a pretrained *generative* teacher. Its final spatial loss does use a pretrained frozen vision encoder, a distinct source of supervision.

**Consistency and one-step flow maps.** Consistency Models learn common endpoints along probability-flow trajectories and can be trained either by distillation or directly from data (Song et al., 2023; Song & Dhariwal, 2024); Consistency Trajectory Models extend the map to intermediate destinations (Kim et al., 2024). Shortcut Models and Flow Map Matching learn finite transitions (Frans et al., 2025; Boffi et al., 2025). MeanFlow instead models average velocity, with iMF refining the regression formulation and guidance (Geng et al., 2025; 2026; Kim et al., 2026). Pixel MeanFlow separates an image-valued prediction from the space in which its loss is defined (Lu et al., 2026b). BWM likewise predicts RGB directly, but its inherited learning rule comes from stationary terminal-map conservation, with no interpolant-time input to the network.

**Autonomous transport and Drifting.** Equilibrium Matching and Equilibrium Forcing investigate time-independent generation (Wang & Du, 2025; Lillemark et al., 2026). BTM establishes autonomous endpoint transport under explicit support and regularity assumptions, then motivates direct map learning through a conservation equation (Lee et al., 2026); discrete BTM adapts this perspective to simplex dynamics (Tang & Wang, 2026). Drifting constructs a kernel-based attraction–repulsion field from real and generated samples (Deng et al., 2026). DriftWorld estimates a separate field for each history–action condition and already incorporates motion weighting and spatial features (Lu et al., 2026a). BWM uses sampled directional derivatives instead of a conditional negative population, then applies supervised losses to the pure-source endpoint. We compare the resulting implementations, including their training costs.

**World prediction and control.** Action-conditioned prediction and visual foresight connect learned image dynamics to robotic control (Finn et al., 2016; Ebert et al., 2018; Babaeizadeh et al., 2021). Video diffusion, MCVD, and latent video models broaden the predictive representation (Ho et al., 2022; Voleti et al., 2022; He et al., 2023), while latent imagination supports policy learning (Hafner et al., 2020; 2023). GPC, AVDC, Ctrl-World, and WorldGym evaluate models through manipulation, planning, or policy assessment (Qi et al., 2026; Ko et al., 2024; Guo et al., 2026; Quevedo et al., 2025). Concurrent DIDO distills a world action model with explicit interaction guidance (Lyu et al.,

2026). Our study isolates one-call RGB prediction on two manipulation benchmarks, with action sensitivity and downstream utility assessed separately from visual error.

# 3 BECKMANN WORLD MODELS

## 3.1 PROBLEM FORMULATION

Given four observed RGB frames $h$, aligned future controls $a$, and recorded future $y$, let $c = (h, a)$. BWM predicts a native future chunk in one network call:

$$\widehat{y} = T_\theta(z, c) = z + U_\theta(z, c), \qquad z \sim \mathcal{N}(0, 1.35^2 I). \tag{1}$$

The residual U-Net is shared in architecture with the task-matched regression and DriftWorld controls. Images are $96 \times 96$ and normalized to $[-1, 1]$. PushT predicts $F = 4$ frames from two-dimensional controls; Robomimic Can predicts $F = 2$ from seven-dimensional controls. Write $\|v\|_D^2 = D^{-1}\|v\|_2^2$, where $D = 3F96^2$.

## 3.2 PAIR-BASED TERMINAL-MAP LEARNING

For fixed $c$, an autonomous field $b_c$ defines trajectories $\dot{X}_\tau = b_c(X_\tau)$. Under BTM's support and regularity assumptions (Lee et al., 2026), its terminal map is constant along trajectories and fixes the target:

$$T_c(X_\tau(x)) = T_c(x), \qquad J_x T_c(x) b_c(x) = 0, \qquad T_c(y) = y. \tag{2}$$

Transport time is distinct from video time: history and controls remain fixed while the future RGB tensor is transported. We learn this common endpoint directly. For a recorded noise–future pair, sample $s \sim \mathcal{U}[0, 1]$ and construct

$$x_s = (1 - s)z + sy, \quad d = y - z, \quad p = T_\theta(x_s, c), \quad u = \mathrm{sg}[p + 0.25 J_x T_\theta(x_s, c) d]. \tag{3}$$

The JVP differentiates future coordinates only. Its sampled direction estimates the derivative along the interpolant's conditional field without fitting a separate drift network. Stopping $u$ prevents reverse differentiation through the target. With $e_i = \|p_i - u_i\|_D^2$, the stabilized pair objective is

$$\mathcal{L}_{\mathrm{tr}} = \frac{1}{B} \sum_i \frac{e_i}{\mathrm{sg}(e_i) + 0.01}, \quad \mathcal{L}_{\mathrm{bd}} = \mathbb{E}\|T_\theta(y, c) - y\|_D^2, \tag{4}$$

$$\mathcal{L}_{\mathrm{BTM}} = \lambda \mathcal{L}_{\mathrm{tr}} + \mathcal{L}_{\mathrm{bd}} + \mathbb{E}[\mathbf{1}_{s>0.8}\|p - y\|_D^2]. \tag{5}$$

The detached coefficient $\lambda$ balances transport and boundary gradient norms. This pair-based update needs neither a conditional negative population nor a pretrained generative teacher.

**Why learn the terminal map?** A velocity model describes how a sample should move and needs an integration rule to produce a future. Here, the network represents the future endpoint itself. Equation 2 ties intermediate locations to a common destination; Equation 3 turns that constraint into a trainable pair-based target. This distinguishes BWM's learning rule from a regression-only objective even though both deploy one U-Net call. The advantage sought is an accurate terminal prediction at fixed query cost, rather than fewer calls than other one-step models.

## 3.3 SUPERVISING THE DEPLOYED ENDPOINT

The transport objective trains mixed inputs and the data boundary, whereas inference queries pure noise. Good boundary reconstruction alone does not constrain the pure-source prediction: $s = 1$ supplies the answer as input, whereas $s = 0$ contains no future pixels. Uniform interpolation assigns neither endpoint positive probability mass. The observed $27\times$ source-to-boundary error gap motivates an explicit source term instead of relying on interpolation training to generalize to deployment. We close this gap by supervising the deployed prediction:

$$p_0 = T_\theta(z, c), \qquad \mathcal{L}_{\mathrm{src}} = \mathbb{E}\|p_0 - y\|_D^2. \tag{6}$$

For conditionally independent source noise and recorded target, with $P = T_\theta(z, c)$,

$$\mathbb{E}[\|P - y\|_D^2 \mid c] = \|\mathbb{E}[P \mid c] - \mathbb{E}[y \mid c]\|_D^2 + \mathrm{Var}_D(P \mid c) + \mathrm{Var}_D(y \mid c). \tag{7}$$

Thus source supervision penalizes both conditional-mean error and source-induced spread. It is especially useful for paired prediction on these largely deterministic manipulation tasks.

**Action-relevant structure.** All endpoint losses act on the same $p_0$. A target-defined moving mask emphasizes robot and object motion, a temporal term matches within-chunk changes, and a range penalty regularizes RGB values:

$$\mathcal{A}(p, y) = 0.1\mathcal{L}_{\text{move}} + 0.05\mathcal{L}_{\text{temp}} + 0.01\mathcal{L}_{\text{range}}. \tag{8}$$

The final continuation additionally matches channel-normalized relu1_2, relu2_2, and relu3_3 features from frozen ImageNet VGG16. Let $\phi_\ell$ be the three frozen feature maps and $N(v) = v/(\|v\|_2 + 10^{-6})$ normalize each spatial channel vector. The spatial objective is

$$\mathcal{L}_{\text{sp}} = \frac{1}{3}\sum_{\ell=1}^{3} \text{mean}_{b,f,u,v} \left\| N(\phi_\ell(p_0)_{bfuv}) - N(\phi_\ell(y)_{bfuv}) \right\|_2^2 . \tag{9}$$

Target features are detached. Feature alignment complements pixel accuracy by supervising shape and appearance at multiple spatial scales. Reusing $p_0$ adds no world-model forward pass, and the encoder is absent at inference.

**Generated-history recovery.** Every fourth update, a detached model-generated prefix replaces the visual history for a separate recorded endpoint. The aligned controls and target remain recorded. This regularizes autoregressive use without backpropagating through the prefix. For the recovery prediction $p_{\text{rec}} = T_\theta(z_{\text{rec}}, c_{\text{rec}})$, define

$$\mathcal{R} = 0.25 r_k I_k \left[ \|p_{\text{rec}} - y_{\text{rec}}\|_D^2 + \mathcal{A}(p_{\text{rec}}, y_{\text{rec}}) \right], \qquad I_k = \mathbf{1}_{(k+1) \bmod 4 = 0}. \tag{10}$$

The complete objective is

$$\mathcal{L} = \mathcal{L}_{\text{BTM}} + \mathcal{L}_{\text{src}} + r_k \mathcal{A}(p_0, y) + \mathcal{R} + \alpha_k \mathcal{L}_{\text{sp}}. \tag{11}$$

The motion ramp is $r_k = \min(k/1000, 1)$. Spatial supervision uses $\alpha_{k_s} = 0.05 \min(k_s/1000, 1)$, with $k_s$ measured from the spatial continuation; the recovery clock does not restart. Training introduces source, motion/recovery, and spatial terms in successive continuations. Their roles address three concrete mismatches: mixed-input training versus pure-source inference, static-pixel dominance versus action-relevant motion, and real-history training versus generated-history rollout. Exact coefficients, ramps, and tensor alignment are in Appendix A. Algorithm 1 separates the conserved-map update from the endpoint supervision; inference remains Equation 1.

# 4 Experiments

## 4.1 Setup and Prediction Accuracy

We evaluate PushT and single-view Robomimic Can with the released RGB interfaces (Lu et al., 2026a; Mandlekar et al., 2021). The task U-Nets have approximately 8.4M and 74.2M parameters. DriftWorld, MSE U-Net, GPC, and the AVDC-derived control train from scratch for five configured epochs on the same manifests. BWM uses the cumulative continuation in Table 2. The 64-frame benchmark conditions on four real frames and scores 60 generated frames across 388 PushT and 30 Can episodes. We report raw normalized-RGB MSE, target-defined moving-region MSE, and AlexNet LPIPS (Zhang et al., 2018), averaging frames within episodes and then weighting episodes equally. Full-episode evaluation stresses longer horizons. These are development validation cohorts; training budgets differ across complete recipes.

**Algorithm 1** BWM: pair-based terminal-map learning and endpoint alignment

**Require:** Recorded history $h$, controls $a$, future $y$; terminal map $T_\theta$
$c \leftarrow (h, a)$; sample $z \sim \mathcal{N}(0, 1.35^2 I)$ and $s \sim \mathcal{U}[0, 1]$
**Learn the common terminal map:**
  $x_s \leftarrow (1-s)z + sy$; $p \leftarrow T_\theta(x_s, c)$
  $u \leftarrow \mathrm{sg}[p + 0.25 J_x T_\theta(x_s, c)(y - z)]$ ▷ condition fixed
  form $\mathcal{L}_{\mathrm{BTM}}$ from the stopped target and boundary losses (Eq. 5)
**Align the prediction used at deployment:**
  $p_0 \leftarrow T_\theta(z, c)$; add source loss $\|p_0 - y\|_D^2$
  add motion/temporal and frozen spatial-feature losses on $p_0$
optionally add detached-prefix recovery loss; update $\theta$ and EMA
**Inference:** sample $z$ and return $T_\theta(z, (h, a))$ **in one call**

Table 1: **Prediction accuracy and measured inference cost.** The 64-frame benchmark starts from four observed frames and scores 60 autoregressively generated frames. Latency is per native chunk on one A800 at batch size one. Scratch baselines use our held-out split for five configured epochs; BWM uses the longer staged lineage in Table 2. Calls are network evaluations per predicted chunk. Lower is better.

| Task | Method | Calls | ms ↓ | MSE ↓ | Moving ↓ | LPIPS ↓ | Full MSE ↓ |
|---|---|---|---|---|---|---|---|
| Can | DriftWorld | 1 | 23.8 | .0104 | .0769 | .0682 | .0256 |
| | MSE U-Net | 1 | 23.7 | .0051 | .0544 | .0104 | .0138 |
| | GPC, same split | 6 | 70.3 | .0276 | .3103 | .0589 | .0380 |
| | AVDC, same split | 100 | 2374 | .0400 | .2644 | .1194 | .1768 |
| | **BWM** | **1** | 23.9 | **.0028** | **.0298** | **.0073** | **.0114** |
| PushT | DriftWorld | 1 | 21.5 | .0314 | .4093 | .0790 | .0451 |
| | MSE U-Net | 1 | 21.8 | .0303 | .4008 | .0804 | .0427 |
| | GPC, same split | 12 | 140.7 | .0630 | 1.4241 | .1609 | .0753 |
| | AVDC, same split | 100 | 2085.5 | .0684 | 1.1840 | .1860 | .0795 |
| | **BWM** | **1** | **20.6** | **.0155** | **.1933** | .0407 | **.0231** |

BWM reduces 64-frame MSE versus same-backbone regression by $48.8\%$ on PushT and $45.1\%$ on Can, at comparable one-call latency. It also improves MSE, moving-region MSE, and LPIPS over local DriftWorld on both tasks. The qualitative comparisons in Figure 2 show the corresponding object-shape and arm-configuration predictions.

### 4.2 Inference and Training Efficiency

Figure 3 measures synchronized inference on one A800 with resident inputs. BWM takes $20.6$ ms per four-frame PushT chunk and $23.9$ ms per two-frame Can chunk. On PushT, replacing 20-step autonomous integration by the terminal map reduces latency $19.9\times$; BWM is $6.8\times$ faster than GPC and $101.2\times$ faster than the 100-call AVDC control. One-call regression and DriftWorld have comparable latency. Figure 4 includes inherited training time. The source-anchored Can map reaches LPIPS $0.0225$ after $2.64$ hours; the motion-aware continuation reaches $0.0090$ after $4.32$ hours, versus DriftWorld's $0.0682$ after $10.19$ hours. Complete-update profiling measures $3.62$ versus $1.09$ windows/s and $7.77$ versus $36.07$ GiB for the motion-aware implementation and DriftWorld. Different optimizers and exposure make these complete-recipe comparisons; PushT inherits more updates.

### 4.3 Quantitative and Qualitative Recipe Ablations

Source anchoring provides the first large improvement, followed by motion-aware recovery and spatial supervision. From the stabilized map to BWM, 64-frame MSE falls from $0.0501$ to $0.0155$ on PushT and $0.0182$ to $0.0028$ on Can. Spatial continuation improves LPIPS on both tasks; Can full-episode MSE changes from $0.0112$ to $0.0114$. Figure 5 directly compares the motion-aware parent and final spatial model under the same recorded inputs. The final model sharpens the PushT

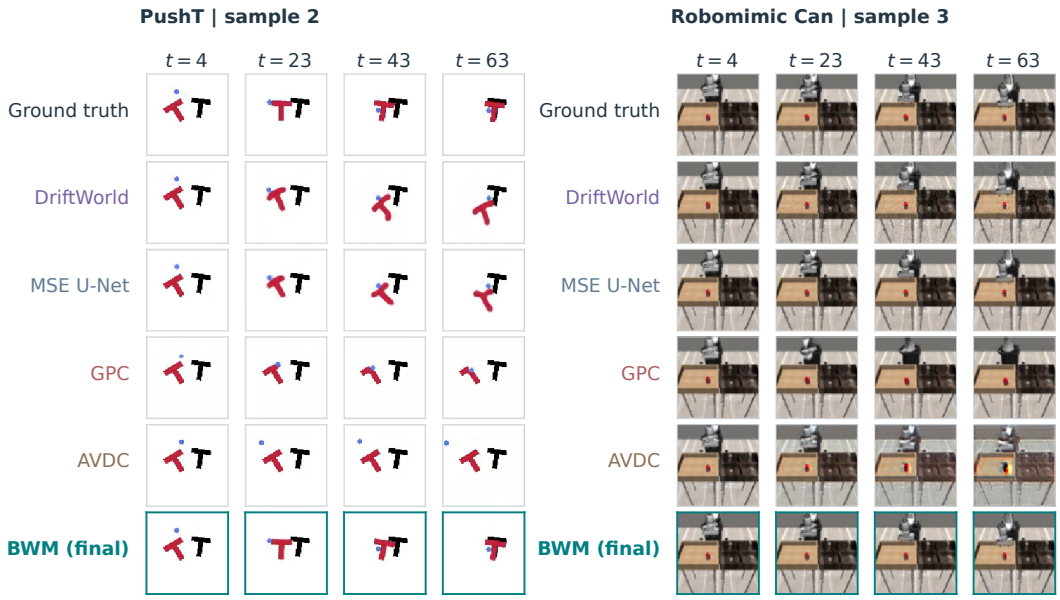



Figure 2: **Selected final-checkpoint predictions against all four baselines.** DriftWorld, MSE U-Net, GPC, AVDC, and BWM share the four-frame real history, recorded actions, and timestamps within each task. All baselines are trained on our split; training budgets differ. These are selected low-error cases. Appendix C gives four examples per task, checkpoint details, selection context, and continuation comparisons.

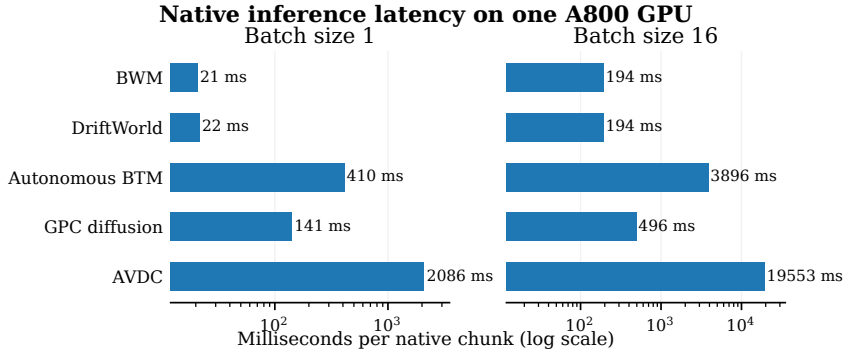



Figure 3: **Native inference latency on one A800.** BWM ("Ours + spatial") uses one network call per chunk; autonomous BTM uses 20, GPC uses 12 on PushT and six on Can, and AVDC uses 100. Timing includes synchronized model inference after warmup.

silhouette and refines Can foreground structure in these selected episodes. These are cumulative recipe comparisons; Table 2 and Appendix C give the quantitative and broader visual context.

**Why source alignment matters.** The stabilized map's source error is $27\times$ its data-boundary error. Source anchoring lowers fixed-window source MSE from $0.00283$ to $0.00082$, and the motion-aware parent reaches $0.00050$. Across eight source draws on 64 windows, BWM reduces moving-region output spread $20.9\times$ versus DriftWorld while retaining action dependence: the action-swap penalty is $0.6447$ versus $0.5339$. These probes support the source-loss decomposition in Equation 7. Figure 6 visualizes the source-alignment ablation.

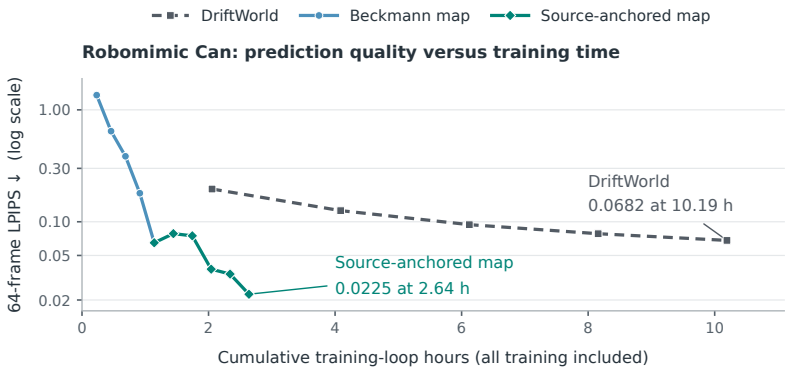



Figure 4: **Training efficiency on Robomimic Can.** LPIPS versus cumulative recorded training-loop hours, including inherited parents. Source anchoring reaches 0.0225 after 2.64 h, versus DriftWorld's 0.0682 after 10.19 h. Markers are evaluated checkpoints; recipes use different optimizers and exposure.

Table 2: **Ablation study.**

| | PushT | | Robomimic Can | |
|---|---|---|---|---|
| Training recipe | LPIPS ↓ | Full MSE ↓ | LPIPS ↓ | Full MSE ↓ |
| Initial BTM port | .8087 | — | — | — |
| Stabilized map | .1119 | .0627 | .0649 | .1159 |
| + source anchoring | .0746 | .0434 | .0225 | .0940 |
| + motion-aware recovery | .0564 | .0299 | .0090 | **.0112** |
| + spatial endpoint, BWM | **.0407** | **.0231** | **.0073** | .0114 |
| Learned direction alternative | .0501 | .0319 | .0854 | .0759 |
| DriftWorld reference | .0790 | .0451 | .0682 | .0256 |




Figure 5: **Visual recipe ablation: motion-aware parent to final spatial BWM.** Same recorded histories/actions and frame 63. The selected PushT and Can examples expose changes in object boundaries and arm appearance. Rows are successive trained continuations with different cumulative budgets, not an equal-budget isolated-loss test.

### 4.4 Downstream Utility

Across seven PushT policies with 300 trials each, BWM reaches Pearson correlation 0.946 and policy-IoU error 0.0098, improving the latter by 19.0% over regression. Native GPC-RANK yields 0.708 versus local DriftWorld's 0.709 (paired difference $-0.001$, 95% bootstrap interval $[-0.055, 0.054]$).

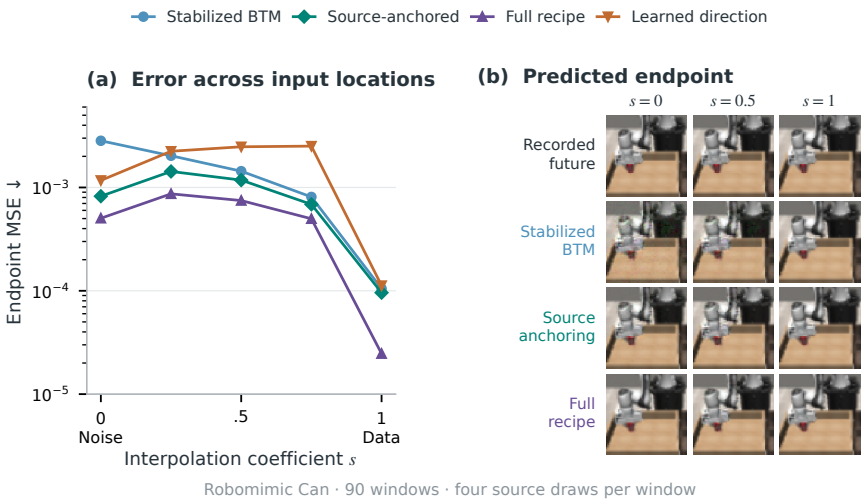



Figure 6: **The source query needs its own supervision.** (a) Independent one-call errors on 90 Can windows with four shared source draws each. Accuracy near recorded data does not ensure accuracy from noise. (b) A fixed crop from the first episode's middle window. Only $s = 0$ is deployment; other inputs contain target information. "Full recipe" here denotes the motion-aware parent before spatial supervision.

Table 3: **PushT downstream evaluation.** Seven policies use 300 trials each. MAE measures absolute policy-IoU error. GPC-RANK evaluates 50 candidates on each of 50 fixed environment seeds.

| World model | Policy Pearson $r$ ↑ | Policy MAE ↓ | GPC-RANK ↑ |
|---|---|---|---|
| Released DriftWorld | .926 | .0203 | – |
| Local DriftWorld | .857 | .0306 | **.709** |
| MSE U-Net | .915 | .0121 | .695 |
| GPC, same split | .366 | .5127 | .615 |
| **BWM** | **.946** | **.0098** | .708 |

Thus the observed gains concern prediction and policy evaluation; the fixed-proposal planning test does not establish a gain.

## 5 Conclusion

Beckmann World Models formulate action-conditioned prediction as learning the endpoint of an autonomous transport. Pair-based terminal-map learning removes the need to generate a negative population or distill a generative teacher, while endpoint-aligned supervision targets the pure-source query that the deployed model actually receives. Source anchoring, motion-aware recovery, and spatial supervision yield a one-call predictor that improves image accuracy and policy-performance estimation on PushT and Robomimic Can. The result is a practical connection between terminal-map conservation and useful visual world prediction: train the deployment endpoint explicitly, preserve action-relevant structure, and retain the original inference interface. Equal-compute attribution and broader stochastic environments remain directions for extending this evidence.

## Reproducibility Statement

Appendix A specifies the architecture, derivatives, losses, and training schedules. Appendix B gives evaluation and inherited training records; Appendix C records sample selection and matched outputs. Measurements come from saved model evaluations. Training repetitions are not included.

## Ethics Statement

This work studies visual prediction for manipulation. The evaluations do not establish suitability for safety-critical robotic control.

## Statement of AI Use

Generative AI tools assisted experiment execution, implementation, debugging, and manuscript preparation. Quantitative results come from model evaluation records, and visualizations use recorded observations and model outputs. The authors are responsible for the final paper.

## REFERENCES


Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Mohammad Babaeizadeh, Mohammad Taghi Saffar, Suraj Nair, Sergey Levine, Chelsea Finn, and Dumitru Erhan. FitVid: Overfitting in pixel-level video prediction. *arXiv preprint arXiv:2106.13195*, 2021.

David Berthelot, Arnaud Autef, Jierui Lin, Dian Ang Yap, Shuangfei Zhai, Siyuan Hu, Daniel Zheng, Walter Talbott, and Eric Gu. TRACT: Denoising diffusion models with transitive closure time-distillation. *arXiv preprint arXiv:2303.04248*, 2023.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *arXiv preprint arXiv:2406.07507v2*, 2025.

Jake Bruce et al. Genie: Generative interactive environments. *arXiv preprint arXiv:2402.15391*, 2024.

Mingyang Deng, He Li, Tianhong Li, Yilun Du, and Kaiming He. Generative modeling via drifting. *arXiv preprint arXiv:2602.04770v2*, 2026.

Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018.

Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems*, 2016. URL `https://arxiv.org/abs/1605.07157`.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations*, 2025.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J. Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models. *arXiv preprint arXiv:2512.02012v2*, 2026.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A controllable generative world model for robot manipulation. In *International Conference on Learning Representations*, 2026.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. *arXiv preprint arXiv:2211.13221v2*, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems*, 2022.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, 2022.

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.02279.

Jin-Young Kim, Hyojun Go, Lea Bogensperger, Julius Erbach, Nikolai Kalischek, Federico Tombari, Konrad Schindler, and Dominik Narnhofer. Understanding, accelerating, and improving meanflow training. In *IEEE and CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B. Tenenbaum. Learning to act from actionless videos through dense correspondences. In *International Conference on Learning Representations*, 2024.

Cheuk-Kit Lee, Florentin Coeurdoux, Yuyuan Chen, Sophia Tang, Peter Potaptchik, Yilun Du, Michael S. Albergo, and Eric Vanden-Eijnden. Beckmann transport models: From autonomous flows to one-step maps. *arXiv preprint arXiv:2608.01692v3*, 2026.

Hansen Jin Lillemark, Alex Rojas, Zachary Novack, Runqian Wang, Yilun Du, Yian Ma, Taylor Berg-Kirkpatrick, and Rose Yu. Equilibrium forcing: Adaptive video generation without noise conditioning. *arXiv preprint arXiv:2608.14706*, 2026.

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*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

Susie Lu, Haonan Chen, Weirui Ye, and Yilun Du. DriftWorld: Fast world modeling through drifting. *arXiv preprint arXiv:2607.15065v2*, 2026a.

Yiyang Lu, Susie Lu, Qiao Sun, Hanhong Zhao, Zhicheng Jiang, Xianbang Wang, Tianhong Li, Zhengyang Geng, and Kaiming He. One-step latent-free image generation with pixel mean flows. *arXiv preprint arXiv:2601.22158v3*, 2026b.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

Jing Lyu, Shuanghao Bai, Runze Xiao, Zhenyu Liao, Wenxing Tan, Zihan Tang, Ruochuan Shi, Cheng Peng, Yuheng Ji, Yihao Wang, Badong Chen, Pengwei Wang, Zhongyuan Wang, and Xiaoguang Zhao. DIDO: Distilling interaction-centric dynamics into one-step denoising for world action models. *arXiv preprint arXiv:2609.15570*, 2026.

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 *Conference on Robot Learning*, 2021.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE and CVF International Conference on Computer Vision*, 2023. URL https://arxiv.org/abs/2212.09748.

Han Qi, Haocheng Yin, Aris Zhu, Yilun Du, and Heng Yang. Inference-time enhancement of generative robot policies via predictive world modeling. *IEEE Robotics and Automation Letters*, 2026.

Julian Quevedo, Ansh Kumar Sharma, Yixiang Sun, Varad Suryavanshi, Percy Liang, and Sherry Yang. WorldGym: World model as an environment for policy evaluation. *arXiv preprint arXiv:2506.00613*, 2025.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE and CVF Conference on Computer Vision and Pattern Recognition*, 2022. URL https://arxiv.org/abs/2112.10752.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, 2024.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://arxiv.org/abs/2010.02502.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.14189.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. URL https://arxiv.org/abs/2011.13456.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, 2023.

Sophia Tang and Shiyi Wang. Discrete Beckmann transport models for one-step language modeling and reasoning. *arXiv preprint arXiv:2609.15903*, 2026.

Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. MCVD: Masked conditional video diffusion for prediction, generation, and interpolation. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2205.09853.

Runqian Wang and Yilun Du. Equilibrium matching: Generative modeling with implicit energy-based models. *arXiv preprint arXiv:2510.02300*, 2025.

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations*, 2024.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in Neural Information Processing Systems*, 2024a.

Tianwei Yin, Micha"el Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *IEEE and CVF Conference on Computer Vision and Pattern Recognition*, 2024b.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. *arXiv preprint arXiv:2504.02792*, 2025a.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. IRASim: A fine-grained world model for robot manipulation. In *IEEE and CVF International Conference on Computer Vision*, pp. 9834–9844, 2025b.

# A Implementation Details

## A.1 Tensor Shapes and Action Alignment

History has shape $[B, 3, 4, 96, 96]$. The source and target have shape $[B, 3, F, 96, 96]$, with $F = 4$ for PushT and $F = 2$ for Robomimic Can. Actions have shape $[B, F, A]$, with $A = 2$ and $A = 7$, respectively. The history is flattened into 12 channels and broadcast over future positions. It is concatenated with the three transported RGB channels. Frame-aligned action embeddings condition the residual blocks. The backbone combines spatial and temporal convolutions with spatial attention. It is not a temporal-attention transformer.

For a PushT window using observed images at indices 0 through 3, future images are at indices 4 through 7 and controls are at indices 3 through 6. Absolute planar actions are normalized by $\frac{2a}{511} - 1$. Images are normalized to $[-1, 1]$. A three-frame planning tail is evaluated at its actual length. Padding future actions and then discarding frames could change valid outputs because the inherited temporal mixing is not strictly causal.

The residual map is $T_\theta(x, c) = x + U_\theta(x, c)$. The input tensor differentiated by the JVP is $x$ only. Both $h$ and $a$ remain fixed in its closure. The implementation preserves the forward-mode derivative even though the resulting target is stopped for reverse-mode differentiation. Paths incompatible with forward-mode differentiation are bypassed rather than replaced with an approximate finite difference. This distinction matters for execution cost.

## A.2 Gradient Balancing and Optimization

Let $w$ denote parameters of the output spatial projection. Define

$$g_{\mathrm{tr}} = \nabla_w \mathcal{L}_{\mathrm{tr}}, \qquad g_{\mathrm{bd}} = \nabla_w (\mathcal{L}_{\mathrm{bd}} + \mathcal{L}_{\mathrm{near}}). \tag{12}$$

We average each gradient vector across distributed ranks before taking its Euclidean norm. When both norms exceed $10^{-8}$, the detached balance is

$$\lambda = \mathrm{sg}\left( \mathrm{clip}\left( \frac{\|g_{\mathrm{bd}}\|_2}{\|g_{\mathrm{tr}}\|_2}, 10^{-3}, 10^{3} \right) \right). \tag{13}$$

When either finite norm is at most $10^{-8}$, the coefficient is one. Nonfinite gradients cause a failure rather than a silent substitution. The fallback prevents an initially zero boundary gradient from disabling transport. Source and endpoint auxiliary terms are not inserted into this original balancing group.

The stabilized map uses Muon for hidden weights and AdamW for output and auxiliary parameters. The original target learning rates are $10^{-3}$ and $10^{-4}$. The completed motion-aware continuation uses one quarter of these rates. Warmup lasts 500 updates from one percent of the target rate. The global gradient norm is clipped at two, and EMA decay is $0.999$. Training and the recorded full-update comparison use full precision. Each continuation explicitly initializes from its parent's EMA and resets the continuation's optimizer and scheduler. Resuming the same continuation instead restores its own model, optimizer, scheduler, EMA, RNG states, sampler position, and update clock.

The matched DriftWorld backbone does not imply identical optimization. Its recorded runs use AdamW, smaller window batches, and 16 generated negatives per PushT condition or 32 per Robomimic Can condition. Kernel temperatures are $0.02$, $0.05$, and $0.2$. These differences are part of the complete implementation comparison and prevent attributing every speed or quality difference to a single operation.

## A.3 Exact Endpoint Regularizers

For example $i$, let $y_{i,0}$ be the last recorded image before the target chunk. The target-defined moving mask is

$$m_{i,f,u,v} = \mathbf{1}\left[ \max_{r \in \{1,2,3\}} |y_{i,r,f,u,v} - y_{i,r,0,u,v}| > 0.05 \right]. \tag{14}$$

The RGB channel index is $r$, future position is $f$, and spatial coordinates are $(u, v)$. Broadcast the mask over channels and let $\mathrm{mean}$ average the indicated tensor coordinates. The training loss is

$$\mathcal{L}_{\mathrm{move}} = \frac{1}{B} \sum_{i} \frac{\mathrm{mean}\left[m_i (p_i - y_i)^2\right]}{\max(\mathrm{mean}(m_i), 0.05)}. \tag{15}$$

The floor limits the additional per-coordinate weight induced by the coefficient $0.1$ to at most two. The other terms are

$$\mathcal{L}_{\mathrm{temp}} = \mathrm{mean}_{i,r,f=2:F,u,v} \left[ \left( (p_{i,r,f,u,v} - p_{i,r,f-1,u,v}) - (y_{i,r,f,u,v} - y_{i,r,f-1,u,v}) \right)^2 \right], \tag{16}$$

$$\mathcal{L}_{\mathrm{range}} = \mathrm{mean}\left[\max(|p| - 1, 0)^2\right]. \tag{17}$$

The temporal term concerns within-chunk differences. It does not force a repaired endpoint to preserve the offset of an erroneous generated preceding frame.

The recovery branch uses a separate valid 16-frame training sequence. Robomimic Can generates between one and seven two-frame prefix chunks before a two-frame endpoint. PushT generates between one and three four-frame prefix chunks and alternates four-frame and three-frame endpoints. The depth curriculum expands over 1,000 completed updates. Prefix generation is detached, raw, and conditioned on aligned recorded actions. No blend with the true history or inference clamp is introduced. The branch is active every fourth update, with weight $0.25$ on active updates. It is not multiplied by four to compensate for its frequency. All losses use one final backward pass and one optimizer update.

The clean-window dataset retains short episodes that cannot support a 16-frame sequence. The recovery sequence is obtained from a separate training-only valid-sequence index. No sample crosses an episode boundary. This regularizer does not create labeled counterfactual simulator transitions.

### A.4 Frozen Spatial Endpoint Features

Spatial supervision uses ImageNet-pretrained VGG16 convolutional weights (Simonyan & Zisserman, 2015) and no classifier. Each predicted video tensor is reshaped from $[B, 3, F, 96, 96]$ to $[BF, 3, 96, 96]$, mapped from $[-1, 1]$ to $[0, 1]$, and normalized by the standard ImageNet channel statistics. We extract relu1_2, relu2_2, and relu3_3 at their native spatial resolutions. For feature tensor $v \in \mathbb{R}^{C \times U \times V}$, channel normalization is

$$\bar{v}_{:,u,v} = \frac{v_{:,u,v}}{\|v_{:,u,v}\|_2 + 10^{-6}}. \tag{18}$$

The loss at each level is the mean squared Euclidean distance between normalized channel vectors, summed over channels and averaged over batch, frame, and spatial locations. The three level losses receive equal weight. Target features are computed without gradients. Predictions are not clipped before feature extraction, so out-of-range outputs still contribute gradients.

The spatial coefficient ramps linearly from zero to $0.05$ over 1,000 child updates. The VGG module is frozen and is stored outside the world model, optimizer, EMA, and checkpoint state. The source prediction already required by $\mathcal{L}_{\mathrm{src}}$ is reused, so spatial supervision introduces no additional world-model forward pass. Training fails if the declared feature asset or checksum is unavailable rather than silently substituting random features. At inference the feature module is absent.

**Continuation schedule.** The base map first learns $\mathcal{L}_{\mathrm{BTM}}$. Source supervision ramps to one over 500 updates. Motion/recovery terms ramp over 1,000 updates from the source-anchored EMA. The final spatial stage continues the motion-aware EMA, ramps only its feature coefficient, and retains the parent recovery clock.

## B Evaluation and Training Records

**Evaluation.** The 64-frame cohorts contain 388 PushT and 30 Can episodes; full-episode evaluation uses 400 and 30. Models receive four real frames and recorded controls, then feed back raw predictions. MSE uses normalized RGB without clipping; LPIPS uses clipped RGB. PSNR and SSIM follow the clipped, quantized image convention. The moving mask selects target pixels differing from the last observed frame by more than $0.05$ in any channel. Metrics average within episodes and then equally across episodes. Evaluations use EMA weights and seed 314. Training follows one trajectory per recipe, so uncertainty across episodes is distinct from training-seed uncertainty.

**Training lineage.** Can uses 5,000 updates per stabilized, source, motion/recovery, and spatial stage (20,000 cumulative). PushT cumulative counts are approximately 141,700, 163,300, 183,000, and 202,700; local DriftWorld uses 39,350 PushT and 20,000 Can updates. The learned-direction branch starts from the source parent. The spatial comparison is a continuation study. Training-time curves sum logged training intervals across all parents, excluding evaluation waits. Two A800 GPUs are used for training.

**Baseline compatibility.** Scratch GPC and AVDC use the same split and recorded controls. GPC uses three denoiser evaluations per frame; the AVDC-derived control uses 100 per native chunk. Released checkpoints below have different training provenance and do not establish held-out generalization on this split. One-call methods share comparable inference cost. Full-episode errors and released references are reported separately from selected image diagnostics.

Table 4: Paired differences from DriftWorld on Robomimic Can. Negative values favor the listed map. Wins count episodes with strictly lower error out of 30. Intervals are 95% episode-bootstrap intervals for fixed checkpoints.

| Method | Horizon | Metric | Difference | 95% interval | Wins |
|---|---|---|---|---|---|
| Source-anchored | 64 frames | LPIPS | $-0.0456$ | $[-0.0481, -0.0432]$ | 30 |
| Source-anchored | 64 frames | Moving MSE | $-0.0193$ | $[-0.0293, -0.0090]$ | 21 |
| Source-anchored | 64 frames | MSE | $-0.0035$ | $[-0.0046, -0.0025]$ | 25 |
| Source-anchored | Full episode | MSE | $+0.0684$ | $[+0.0200, +0.1309]$ | 12 |
| Source-anchored | Full episode | LPIPS | $-0.0073$ | $[-0.0306, +0.0198]$ | 21 |
| Motion-aware (ours) | 64 frames | LPIPS | $-0.0592$ | $[-0.0615, -0.0568]$ | 30 |
| Motion-aware (ours) | 64 frames | Moving MSE | $-0.0453$ | $[-0.0529, -0.0378]$ | 29 |
| Motion-aware (ours) | 64 frames | MSE | $-0.0073$ | $[-0.0082, -0.0064]$ | 30 |
| Motion-aware (ours) | Full episode | MSE | $-0.0144$ | $[-0.0160, -0.0128]$ | 30 |
| Motion-aware (ours) | Full episode | LPIPS | $-0.1129$ | $[-0.1218, -0.1043]$ | 30 |

Table 5: **One-call endpoint prediction on Robomimic Can.** Three windows per validation episode are fixed at its first, middle, and last valid starts. Each model predicts two frames from four real history frames. Scores average three windows within each of 30 episodes. Lower is better.

| Predictor | Raw MSE | Moving MSE | LPIPS |
|---|---|---|---|
| Copy last observation | .002407 | .04586 | .00434 |
| DriftWorld | .001473 | .01522 | .00923 |
| Source-anchored | .000811 | .01374 | .00437 |
| Motion-aware (ours) | **.000504** | **.00886** | **.00289** |

### B.1 Complete Prediction Metrics

Table 6: **Complete final 64-frame comparison.** All rows use our common evaluation cohort. Scratch baselines have five configured training epochs; BWM inherits its staged lineage. BWM parent denotes the motion-aware model before spatial endpoint supervision. Released rows have different training provenance and are separated from the same-split models. Moving denotes target-defined moving-region MSE.

| Task | Method | MSE ↓ | Moving ↓ | LPIPS ↓ | SSIM ↑ | PSNR ↑ |
|---|---|---|---|---|---|---|
| PushT | BWM | .0155 | .1933 | .0407 | .964 | 28.64 |
| | BWM parent | .0199 | .2419 | .0564 | .953 | 26.98 |
| | DriftWorld | .0314 | .4093 | .0790 | .927 | 23.78 |
| | MSE U-Net | .0303 | .4008 | .0804 | .943 | 24.18 |
| | GPC diffusion | .0630 | 1.4241 | .1609 | .904 | 19.70 |
| | AVDC-derived | .0684 | 1.1840 | .1860 | .895 | 19.01 |
| | Released DriftWorld | .0085 | .1182 | .0163 | .982 | 32.13 |
| | Released GPC | .0192 | .2446 | .0324 | .966 | 27.00 |
| Can | BWM | .0028 | .0298 | .0073 | .979 | 33.58 |
| | BWM parent | .0031 | .0316 | .0090 | .974 | 33.03 |
| | DriftWorld | .0104 | .0769 | .0682 | .856 | 26.91 |
| | MSE U-Net | .0051 | .0544 | .0104 | .969 | 30.90 |
| | GPC diffusion | .0276 | .3103 | .0589 | .910 | 22.48 |
| | AVDC-derived | .0400 | .2644 | .1194 | .801 | 21.53 |
| | Released DriftWorld | .0525 | .0662 | .0676 | .863 | 19.34 |

Table 7: **Final full-episode comparison.** The same-split and released-checkpoint distinctions of Table 6 apply. A dash denotes an unmeasured quantity. Spatial endpoint supervision does not improve every full-episode metric: the BWM parent has slightly lower Can MSE, and the MSE U-Net has lower Can LPIPS than either BWM variant.

| Task | Method | MSE ↓ | Moving MSE ↓ | LPIPS ↓ |
|---|---|---|---|---|
| PushT | BWM | .0231 | .2451 | .0628 |
| | BWM parent | .0299 | .3074 | .0810 |
| | DriftWorld | .0451 | .4624 | .1115 |
| | MSE U-Net | .0427 | .4512 | .1104 |
| | GPC diffusion | .0753 | 1.4597 | .2135 |
| | AVDC-derived | .0795 | 1.1297 | .2355 |
| | Released DriftWorld | .0139 | .1560 | .0270 |
| | Released GPC | – | – | – |
| Can | BWM | .0114 | .0655 | .0353 |
| | BWM parent | .0112 | .0659 | .0334 |
| | DriftWorld | .0256 | .1316 | .1463 |
| | MSE U-Net | .0138 | .1004 | .0314 |
| | GPC diffusion | .0380 | .3131 | .1257 |
| | AVDC-derived | .1768 | .4040 | .3164 |
| | Released DriftWorld | .0813 | .1287 | .1649 |

## C Qualitative Protocol and Matched Predictions

Four PushT and four Can episodes were selected by the lowest final-BWM decoded RGB MSE over frames 4–63 among eight fixed saved episodes per task. Moving-region ranking selects the same four. This is post-hoc success-case selection, with one saved realization per episode. Histories, controls, and physical timestamps match across methods; archived generator states are not asserted to match. All displayed crops are unretouched $96 \times 96$ outputs. MP4 clipping/compression is inherited, except the final Can AVDC row, exported from its step-2,500 EMA with seed 314, batch eight, and 100 denoiser calls per two-frame chunk. No benchmark metric is recomputed from these selected images.

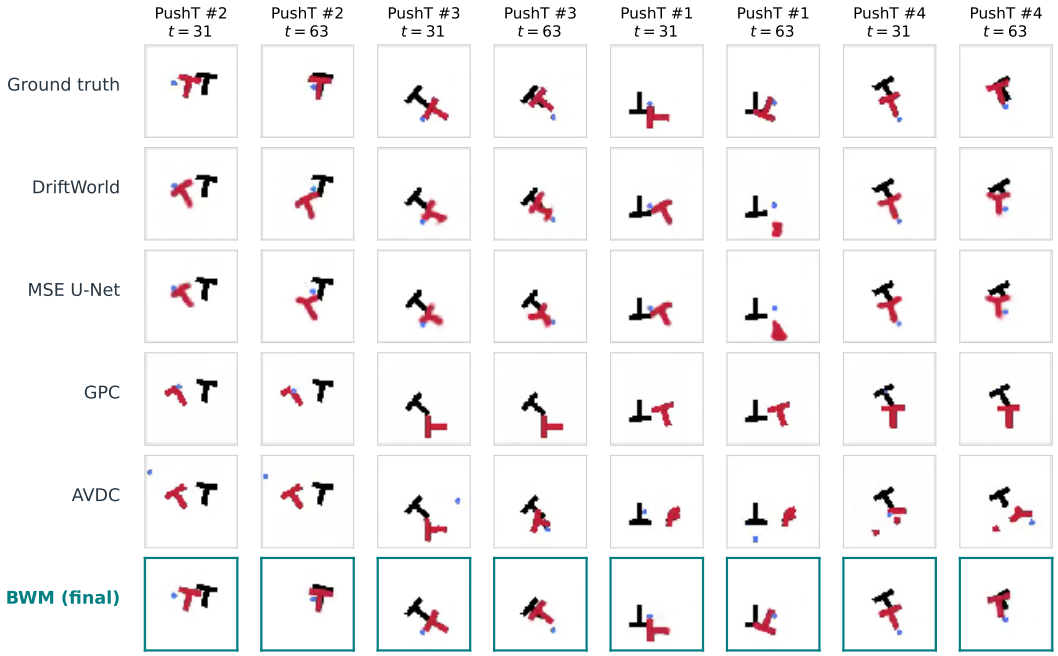



Figure 7: **Four matched PushT episodes.** Two fixed times per saved sample. IDs 2, 3, 1, 4 correspond to `domain18_random_std_10.zarr::50`, `domain18_random_std_20.zarr::204`, `domain18.zarr::10`, and `domain18_random_std_30_v2.zarr::262`.

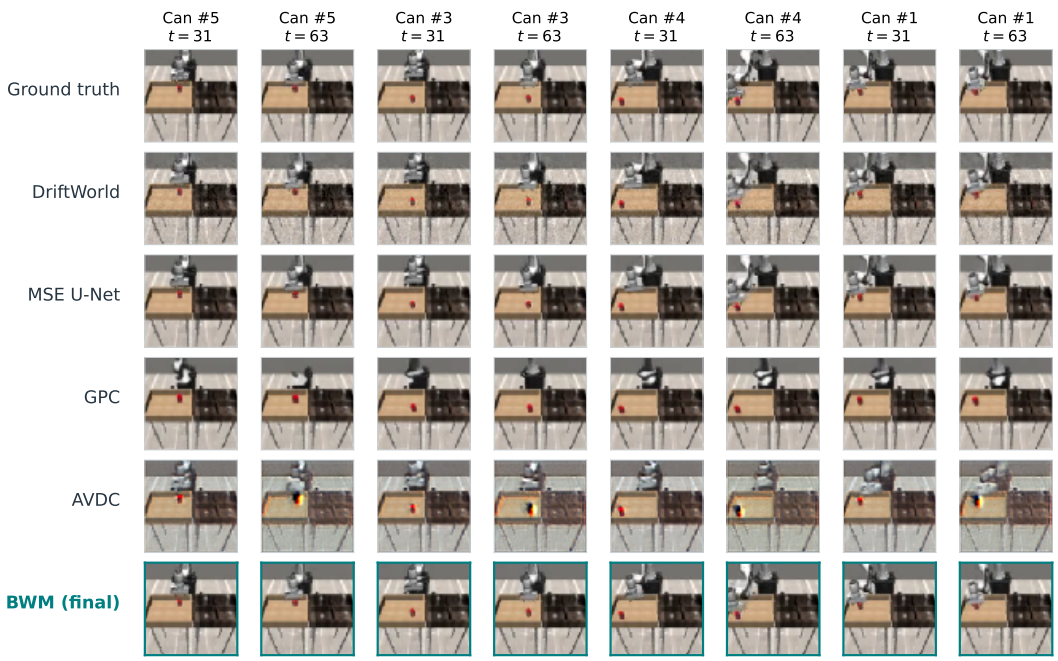



Figure 8: **Four matched Robomimic Can episodes.** Saved IDs 5, 3, 4, 1 are `demo_156`, `demo_62`, `demo_107`, and `demo_4` in `image_v15.hdf5`. Columns compare the same recorded times. These predictions show foreground configuration and appearance, rather than a task-success measurement.

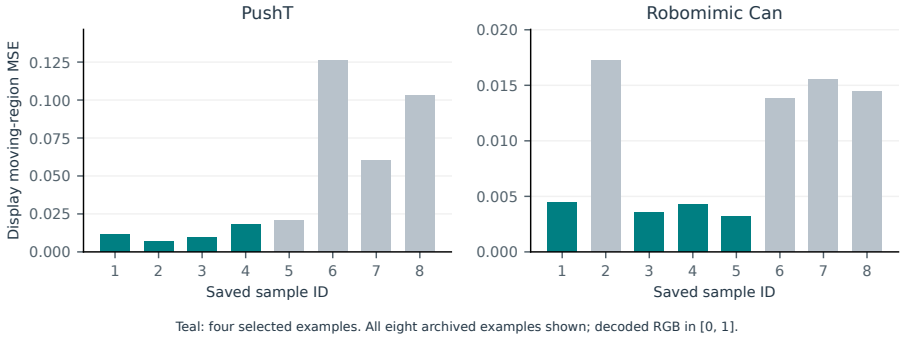



Figure 9: **Selection context.** Display-space moving-region MSE for all eight saved episodes; teal identifies the four selected per task.

## D VISUAL RECIPE COMPARISONS

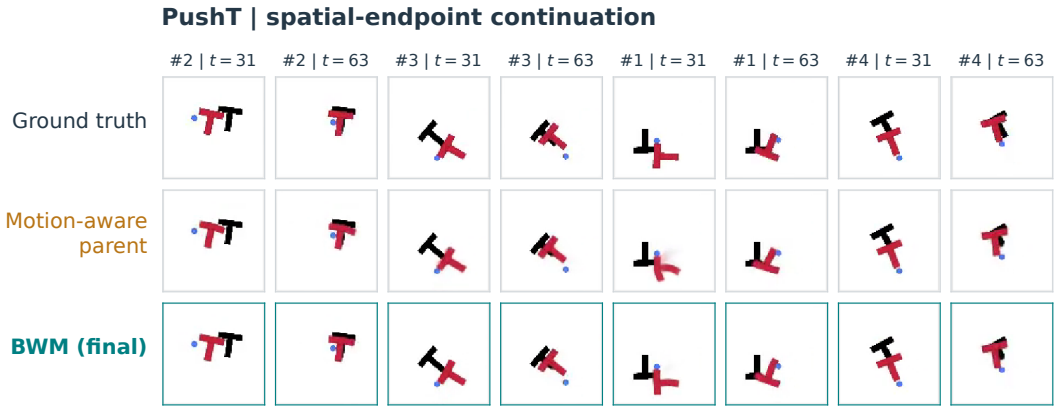



Figure 10: **PushT: motion-aware parent and spatial continuation.** Four selected episodes, each at frames 31 and 63. Final supervision improves edge structure in several cases, with remaining pose error in sample 4.

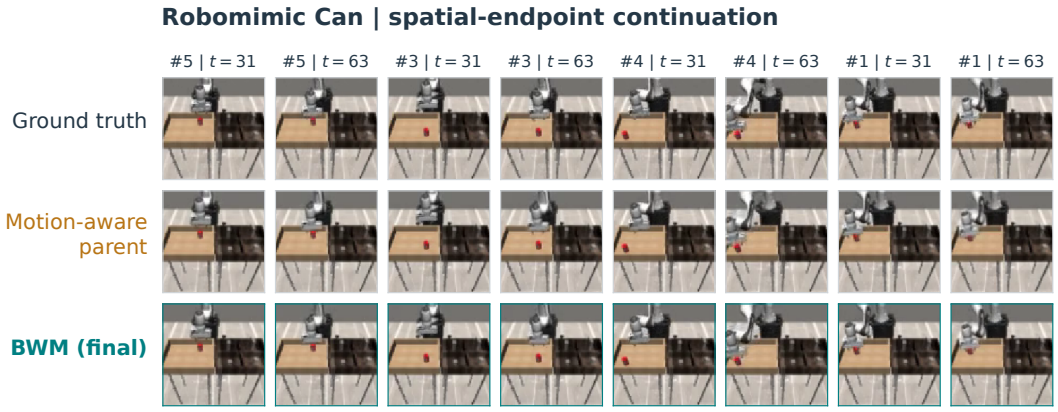



Figure 11: **Can: motion-aware parent and spatial continuation.** Same recorded inputs and times for all four selected episodes. Differences concentrate around the arm and gripper.

## E Endpoint and Action Diagnostics

Fixed-window probes hold recorded history constant. Source spread averages output variance over eight noise draws on 64 windows. Action swap substitutes controls from another window. The source-endpoint probe separately evaluates pure-noise and boundary inputs, distinguishing deployment accuracy from reconstruction. These measurements are native-chunk diagnostics, distinct from the autoregressive benchmark.

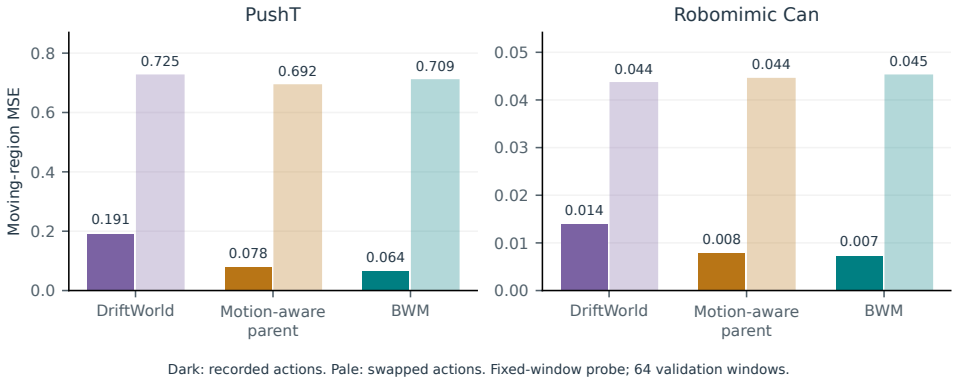



Figure 12: **Action sensitivity across the cumulative recipe.** Recorded-action errors fall while swapped-action errors remain substantially larger. Dark/pale bars denote recorded/swapped actions on 64 windows per task.

**Downstream protocol.** Seven released PushT policies each use 300 model/simulator trials. GPC-RANK evaluates 50 proposed sequences on each of 50 environment seeds with frozen outcome readers. Better paired image prediction need not change ranking within that proposal distribution. The study covers two low-resolution manipulation tasks and does not establish calibrated multimodal coverage or broad video-generation generalization.