# WORLD-MODEL PREDICTIVE DISPLAYS FOR TELEOPERATION

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

## ABSTRACT

Network latency degrades precision in teleoperation, and becomes critical in domains such as remote surgery. Predictive displays show an estimate of the present instead of the delayed camera feed, but learned image-space predictors fix the prediction horizon at training time and are valid only at that delay. We instead build a predictive display around an interactive world model, reaching the present by repeated single-step prediction from the most recent received frame. Building on DINO-WM, we predict in the feature space of a frozen encoder and decode only the displayed frame. A display thread performs one predictor and one decoder evaluation per frame regardless of the delay, while an asynchronous grounding thread re-anchors the latent context to newly arrived observations, so the compensated delay becomes a runtime quantity and jitter requires no retraining. In a within-subjects study with 22 participants on a simulated Push-T task at round-trip delays of 500–1000 ms, spanning the range where fine manipulation breaks down, compensation shortened completion time on successful trials by 23.5% and suppressed the move-and-wait strategy latency induces. Participants rated the display as more responsive but less authentic, and authenticity tracked rollout fidelity.

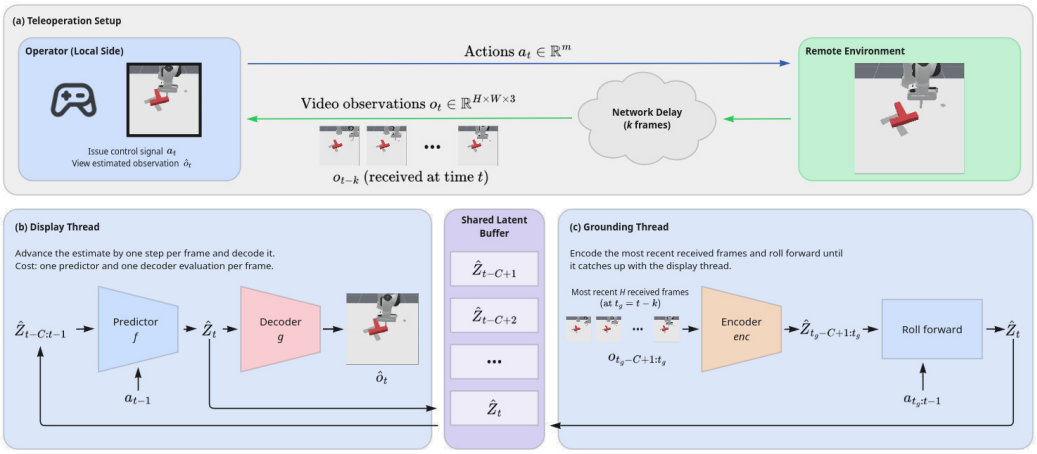


Figure 1: System overview. **(a)** The operator's actions take effect immediately, but video arrives $k$ frames late, so at time $t$ the newest observation is $\boldsymbol{o}_{t-k}$. **(b)** The display thread advances the latent estimate one step and decodes it: one predictor and one decoder evaluation per frame, regardless of $k$. **(c)** The grounding thread asynchronously encodes the newest $C$ received frames, rolls them forward with the stored actions until it catches up, and overwrites the shared latent buffer. The display thread therefore extends a recent ground-truth-anchored estimate instead of recomputing one, keeping the per-frame cost independent of the delay.

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.

# 1 INTRODUCTION

Teleoperation of robotic systems is subject to network latency, which can substantially degrade task performance and increase operator workload. Modern network infrastructure provides low-latency connections in many settings. However, delay remains a problem over long distances, in satellite and wireless links, and in regions with limited infrastructure. These are often exactly the settings where remote operation is most needed, such as providing specialist care to rural or underserved areas, disaster zones, or isolated environments. In domains where precision and safety are paramount, such delays can have serious consequences. In remote surgery in particular, the operator relies on an immediate and faithful view of the system under control. Delay on either the control or the video channel can translate directly into imprecise tool motion and, ultimately, harm to the patient. With direct visual feedback, delays of 200–300 ms already measurably degrade precision and task completion time, and beyond 500 ms fine manipulation becomes nearly unperformable Ferrell & Sheridan (2009); Ferrell (1965); Rosen & Hannaford (2006); Lum et al. (2009). Latency compensation is therefore most needed precisely in the regime where unaided operation has ceased to be viable, which is the range we target in this work.

Latency compensation for teleoperation systems has been widely researched. Especially, passivity-based approaches Hannaford & Ryu (2002); Coelho et al. (2018) or wave variables Anderson & Spong (1988); Niemeyer & Slotine (2002; 2004) can guarantee stability under constant delay by ensuring no energy enters the closed control loop. The trade-off of these kind of approaches however is that teleoperation can feel sluggish and damped, hindering precise manipulation. Moreover, passivity based methods only account for delay on the control channels, ignoring any latency on the video feedback for instance. Another way of addressing latency is to decouple the local and remote sides entirely. Instead of one closed control loop spanning over a potentially delayed network both sides interact with a local model of the counterpart respectively. Only the synchronization of the estimated models is subject to latency and overall task performance depends on the quality of these models Bejczy & Kim (1990); Mitra & Niemeyer (2008). Therefore any network latency is experienced as degradation of the model estimates rather than damping of control. In order to keep these models accurate, traditionally prior approaches restrict prediction to non-pixel state information such as kinematics Bejczy & Kim (1990); Richter et al. (2019); Willaert et al. (2012); Antonsen et al. (2024); Liu et al. (2025) or limited haptic models Mitra & Niemeyer (2008); Sun et al. (2024). Those predictions are then used to render virtual views of the target environment in the form of graphical overlays Bejczy & Kim (1990); Richter et al. (2019) or digital twins Willaert et al. (2012); Antonsen et al. (2024); Liu et al. (2025). Such representations, however, inherently ignore complex interactions between the robot and its environment — particularly in surgical settings, where tissue dynamics and tool–tissue contact cannot be faithfully captured in simulation. Predicting the camera image directly has been investigated using vector-field-based approaches Moniruzzaman et al. (2022b); Sharma et al. (2025), generative adversarial networks (GAN) Yoon et al. (2023); Lee et al. (2024); Ko et al. (2020); Moniruzzaman et al. (2023) and virtual views rendered from predicted 3D scene representations Moniruzzaman et al. (2022a), but these methods remain limited in visual fidelity and generalization or rely on extra sensors.

In contrast, we propose to estimate the present frame with an interactive world model. The model runs locally on the operator side and is continuously re-grounded: it rolls out autoregressively from the latest received ground-truth frame, conditioned on the actions issued since. Reaching the present by repeated single-step prediction, rather than by a single learned jump across the delay, is what gives the approach its properties, and we make three contributions on that basis:

- **The compensated delay becomes a runtime quantity rather than a training choice.** Prior image-space methods learn a fixed mapping from a delayed frame to the present one and are valid only at that delay, typically 500 ms Moniruzzaman et al. (2022b; 2023). We keep the rollout affordable with a two-thread scheme: a display thread performs one predictor and one decoder evaluation per frame regardless of the delay, while an asynchronous grounding thread re-anchors the latent context to the newest received observation. Prediction happens in the feature space of a frozen DINOv2 encoder, and only the displayed frame is decoded to pixels.
- **An operator-in-the-loop evaluation.** In a within-subjects study with 22 participants at round-trip delays of 500–1000 ms, we measure the effect on completion time, control strategy, and perceived display quality, and relate all three to rollout fidelity.

- **Support for non-constant delay by construction.** Each grounding pass derives its rollout length from the arrival index of the newest frame, so jitter and frame loss only change the length of individual passes. We evaluate constant delays only and leave an empirical test to future work.

## 2 BACKGROUND

### 2.1 MODEL MEDIATED TELEOPERATION

Addressing network latency in teleoperation by deploying local models has a long history. Earlier approaches propose limited models that predict simpler non-visual state information, but there is an increasing line of work that operates on the image stream directly. Vector-field based approaches Moniruzzaman et al. (2022b); Sharma et al. (2025) predict warp fields to warp past frames into present ones. This retains visual quality better than approaches that synthesize frames directly. However, vector fields can only move pixels that are already present and thus often fail for regions that are not visible in the reference frame, resulting in visual artifacts.

GANs on the other hand generate the present frame directly Yoon et al. (2023); Lee et al. (2024); Ko et al. (2020); Moniruzzaman et al. (2023). Conditioning and reconstruction losses fix the coarse content of the output, but the remaining detail is supplied by the adversarial objective, which rewards realism without regard to the true system state. Thus the generated frames might look plausible but introduce hallucinations.

Moreover, the learned methods lack the flexibility to account for varying latency scenarios. The prediction horizon is fixed at training time, making the model only viable for one specific constant delay case. We propose to use a world model to regress future frames directly by rolling out autoregressively until the present timestep. This way the frame can be generated for any target horizon as long as the visual quality does not degrade.

### 2.2 WORLD MODEL

World models (WMs) are action-conditioned state predictors, modeling $p(\boldsymbol{s}_{t+1} \mid \boldsymbol{s}_{t-C+1:t}, \boldsymbol{a}_t)$. Most commonly $\boldsymbol{s}$ are frames or latent representations of such, and $\boldsymbol{a}$ are low-level robot commands, discrete control inputs, or latent actions inferred from unlabeled video. WMs have been widely proposed for solving robot control tasks by serving as an offline data generator or online training environment Ha & Schmidhuber (2018); Hafner et al. (2023; 2025), as an auxiliary prediction objective for policy learning Ye et al. (2026); Kim et al. (2026); Li et al. (2026), or as a model for test-time planning Finn & Levine (2017); Ebert et al. (2018); Hafner et al. (2019); Zhou et al. (2024).

Besides that, WMs have also been used as interactive simulators allowing real-time control of video game environments Decart et al. (2024); He et al. (2025); Wang et al. (2026) or robot systems Gao et al. (2026). The challenge there is to make the model fast enough for interactive use while retaining sufficient image quality. Diffusion-based models produce visually sharp frames but require multiple denoising passes per frame unless distilled. An insight from WMs used for downstream control is that compressed latent representations are often sufficient for task performance, which permits prediction at much lower cost than pixel-space rollout.

DINO-WM Zhou et al. (2024) follows this approach and predicts in the feature space of a frozen DINOv2 encoder, with no pixel-level reconstruction term in its objective. The encoder maps an RGB observation $\boldsymbol{o}_t$ to patch-level latent features $\boldsymbol{Z}_t \in \mathbb{R}^{n \times d}$, where $n$ is the number of patches and $d$ the feature dimension,

$$\boldsymbol{Z}_t = \mathrm{enc}(\boldsymbol{o}_t), \tag{1}$$

from which the transition model rolls out autoregressively,

$$\hat{\boldsymbol{Z}}_{t+1} = f_\phi(\boldsymbol{Z}_{t-C+1:t}, \boldsymbol{a}_{t-C+1:t})\,. \tag{2}$$

A decoder is trained separately to map latents back to RGB,

$$\hat{\boldsymbol{o}}_t = q_\psi(\boldsymbol{Z}_t). \tag{3}$$

# 3 METHODS

**Setup** We consider a teleoperation system with two channels: the operator issues control signals $\boldsymbol{a} \in \mathbb{R}^m$ and receives video feedback $\boldsymbol{o} \in \mathbb{R}^{H \times W \times 3}$ from the remote side. The video stream is subject to a round-trip delay of $k$ frames. Commands take effect on the remote side at the time they are issued and $\boldsymbol{a}_t$ denotes the command applied during $[t, t+1)$. At wall-clock step $t$ the local side therefore holds the full action history $\boldsymbol{a}_{0:t}$ but only the stale observation $\boldsymbol{o}_{t-k}$. Our goal is to present the operator with an estimate $\hat{\boldsymbol{o}}_t$ of the current remote state instead of $\boldsymbol{o}_{t-k}$.

**Overview** We use a DINO-WM to roll forward from the most recent received frame. A direct implementation would, at every step, encode the newest observation and roll it forward $k$ steps to the present. This is not feasible at display rate: a $k$-step rollout costs $k$ predictor evaluations, which for realistic delays exceeds the frame period. We therefore split the computation across two threads. A display thread advances the estimate by a single step per frame and decodes it, at a cost of one predictor and one decoder evaluation. A grounding thread runs concurrently, starting from the newest received frame and rolling forward until it catches up with the display thread, at which point it overwrites the shared latent context with its result. The display thread thus extends a recent, ground-truth-anchored estimate rather than recomputing one from scratch, and the decoder — which is the more expensive component — is evaluated exactly once per displayed frame and never during grounding.

**Grounding** At startup we wait until $C$ frames have arrived and run one grounding pass synchronously to initialize the shared latent buffer. Thereafter, each grounding pass encodes the most recent $C$ received frames, $\hat{\boldsymbol{Z}}_{t_g-C+1:t_g} = \mathrm{enc}(\boldsymbol{o}_{t_g-C+1:t_g})$ with $t_g = t - k$, and rolls the predictor forward using the stored actions until $t_g$ reaches the current display index. The full context window is then written back under lock, so the display thread always predicts from a window that is internally consistent rather than a mixture of grounded and ungrounded latents.

**Feasibility and effective horizon** Let $\tau_p$ and $\tau_q$ denote predictor and decoder latency and $\tau_f$ the frame period. The display thread requires $\tau_p + \tau_q < \tau_f$. The grounding thread terminates only if it advances faster than wall-clock, i.e. $\tau_p < \tau_f$; under this condition a pass launched at delay $k$ completes after

$$G = \frac{k\tau_p}{\tau_f - \tau_p} \tag{4}$$

frames, which is also the period at which grounding refreshes. A displayed frame is therefore the product of at most $k + G$ ungrounded autoregressive steps. Together these conditions determine the range of delays the system supports on given hardware, and they motivate predicting in latent space: $\tau_p$ is what has to be kept below $\tau_f$. In our experiments on an RTX 2080 Ti, $\tau_p = 6\,\mathrm{ms}$ and $\tau_q = 13\,\mathrm{ms}$ at $\tau_f = 50\,\mathrm{ms}$, yielding $k + G$ of approximately 11, 17 and 23 steps at the delays evaluated in Sec. 4.

**Variable delay** Although we analyze a constant $k$, nothing in the procedure depends on it. The rollout length of each grounding pass is determined at runtime by the arrival index of the newest frame, so jitter lengthens or shortens individual passes, and a lost frame simply means the next pass starts from an older reference. No component of the model is parameterized by the delay, and no retraining is required to operate at a different one.

# 4 EXPERIMENTS

## 4.1 SETUP

**Task** We evaluate our system on the Push-T task in the ManiSkill simulation framework. The end effector's orientation and its translation along the $z$-axis are held fixed, and actions are restricted to a constant magnitude or zero. The action space thus reduces to the set of 2D unit vectors together with the zero vector, with sampled directions rescaled to the desired end-effector speed. This restriction narrows the gap between the action distribution seen during training and the one produced by an

**Algorithm 1** World Model based Latency Compensation

**Require:** Encoder enc, predictor $f_\phi$, decoder $q_\psi$, context length $C$, frame period $\tau_f$
**Require:** Shared latent buffer $\mathcal{Z}$, action buffer $\mathcal{A}$, lock $\mathcal{L}$

**Procedure** DISPLAYTHREAD
**spawn** INPUTLOGGER() {appends $\boldsymbol{a}_t$ to $\mathcal{A}$ every $\tau_f$}
**wait until** $C$ frames received
$t \leftarrow$ GROUND() {synchronous warm-up}
job $\leftarrow$ **null**
**while** running **do**
  $t \leftarrow t + 1$
  **with** $\mathcal{L}$: $\boldsymbol{Z}_t \leftarrow f_\phi(\mathcal{Z}[t{-}C{:}t{-}1],\, \mathcal{A}[t{-}C{:}t{-}1])$;  $\mathcal{Z}[t] \leftarrow \boldsymbol{Z}_t$
  $\hat{\boldsymbol{o}}_t \leftarrow q_\psi(\boldsymbol{Z}_t)$
  DISPLAY($\hat{\boldsymbol{o}}_t$)
  **if** job = **null or** TERMINATED(job) **then**
    job $\leftarrow$ **spawn** GROUND()
  **end if**
  SLEEPUNTIL($t \cdot \tau_f$)
**end while**
CANCEL(job)
**End Procedure**

**Procedure** GROUND
$t_g \leftarrow$ index of newest received frame $\{t_g = t - k$, measured at runtime$\}$
$\hat{\mathcal{Z}}[t_g{-}C{+}1{:}t_g] \leftarrow \mathrm{enc}\big(\boldsymbol{o}_{t_g-C+1:t_g}\big)$
**while** $t_g <$ NOW() **do**
  $\hat{\mathcal{Z}}[t_g{+}1] \leftarrow f_\phi\Big(\hat{\mathcal{Z}}[t_g{-}C{+}1{:}t_g],\, \mathcal{A}[t_g{-}C{+}1{:}t_g]\Big)$
  $t_g \leftarrow t_g + 1$
**end while**
**with** $\mathcal{L}$: $\mathcal{Z}[t_g{-}C{+}1{:}t_g] \leftarrow \hat{\mathcal{Z}}[t_g{-}C{+}1{:}t_g]$
**return** $t_g$
**End Procedure**

operator at test time, which the predictor $f_\phi$ is otherwise sensitive to. Teleoperation is performed with a single joystick on a game controller at a rate of 20 Hz.

**Training** We train the DINO-WM on data generated online rather than from a fixed offline dataset. A task policy is first trained with PPO and then rolled out in $N = 64$ parallel environments to supply training data for the world model. In environment $i$, the policy's action is replaced with a random direction or the idle action with probability $\frac{i}{N-1}$, so that the batch spans a range from purely on-policy to fully random behavior. Policy-driven rollouts keep the trajectories task-relevant — in particular, they ensure that contact with the T-block is represented in the data — while the injected noise provides the off-manifold coverage the model needs to remain stable under operator commands that no policy would produce. The encoder operates on $224 \times 224$ frames, and the predictor uses a context length of $C = 3$. Predictor and decoder are trained concurrently on separate objectives for 80,000 steps, with the encoder kept frozen (Appendix A).

**User study** We evaluated our system in a within-subjects study with 22 participants, who controlled the task with a gamepad at three round-trip delays (500, 750, and 1000 ms, i.e. $k = 10, 15, 20$), each performed both with and without world-model compensation. Participants completed 10 undelayed practice trials followed by 24 measured trials (4 per condition), capped at 60 s, with condition order counterbalanced across participants. Completion times are reported in simulated task time; the wall-clock cost of inference and the resulting variation in censoring times are detailed in Appendix B.8. Alongside completion time we recorded control-behaviour and prediction-fidelity metrics, the latter computed against the simulator's ground-truth frames at the displayed time step. Subjective ratings were collected after each condition using selected NASA-TLX subscales

(mental demand, performance, frustration) complemented by custom items for perceived responsiveness and display authenticity.

### 4.2 RESULTS

**Completion time** Our system shortened completion time relative to the baseline (Fig. 2, Table 1). In a Cox model stratified by delay, with trials reaching the 60 s cap treated as censored, the instantaneous completion rate was about 30% higher under compensation (HR $= 1.30$, 95% CI $[1.04, 1.62]$, $p = .024$). On successful trials, compensation reduced completion time by 23.5% ($t(21) = -6.35$, $p < .001$, paired on participant means), with 20 of 22 participants faster under compensation. Completion slowed substantially as delay increased: median completion time rose from 27.1 to 32.4 to 37.2 s without compensation and from 20.5 to 23.7 to 29.0 s with it. The reduction was largest at the shortest delay and comparable at the two longer ones (28.0%, 21.3% and 21.6% at 500, 750 and 1000 ms; all $p < .01$), and the benefit did not depend reliably on delay ($\chi^2(2) = 1.40$, $p = .496$), despite the growth of the ungrounded horizon $k+G$ and the fidelity degradation reported in Sec. 4.2. Overall, 24.4% of trials reached the cap, and timeout rates did not differ reliably between conditions (25.0% with compensation vs. 23.9% without, Wilcoxon signed-rank on participant rates, $p = .53$). Compensation therefore made successful trials substantially faster without measurably changing how often the task was completed at all.

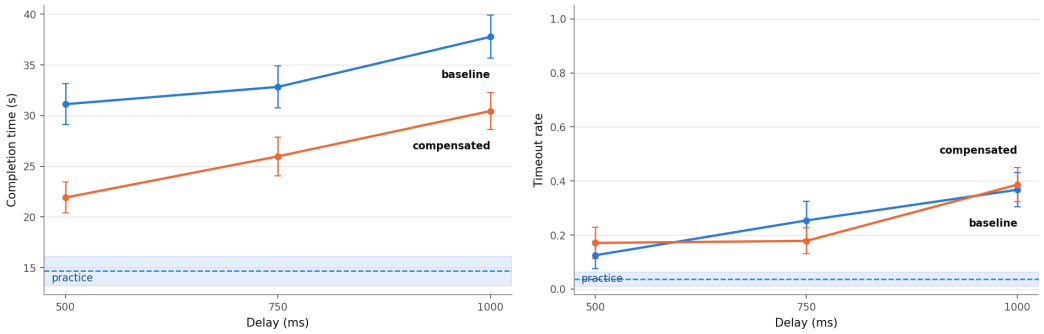


Figure 2: Completion time and timeout rates. **Left:** The predictive display reduced completion time by 23.5% on successful trials. **Right:** The timeout rates increased slightly with the delay but did not differ significantly between the compensated and baseline runs

Table 1: Cox proportional hazards model of completion time in simulated task time, stratified by delay, with robust standard errors clustered by participant and trials reaching the 60 s cap right-censored at their own censoring time. A hazard ratio (HR) above 1 indicates faster completion. Delay is a stratifying variable, so its effect is not estimated as a hazard ratio and is reported descriptively in the text. The interaction, tested jointly across both delay contrasts, was not significant: the benefit of compensation did not depend reliably on delay and held across the range tested.

| Effect | Estimate | 95% CI | $p$ |
|---|---|---|---|
| compensated vs baseline | HR = 1.296 | [1.035, 1.622] | .024 |
| condition × delay | $\chi^2(2) = 1.40$ | — | .496 |

**Control behaviour** The predictive display changed how participants controlled the task, not only how quickly they finished it. Idle fraction — the proportion of each trial during which no control input was issued — fell under compensation in 21 of 22 participants, while input reversal rate rose in all 22 (Fig. 3). The two conditions also diverged across delay: without compensation, idle fraction climbed from 0.60 to 0.72 and reversal rate declined from 0.62 to 0.40 $s^{-1}$ as latency grew (Friedman tests, both $p < .001$), whereas under compensation neither changed reliably (idle fraction 0.43 to 0.47, $p = .24$; reversal rate 0.89 to 0.83 $s^{-1}$, $p = .81$). Together these measures describe the move-and-wait strategy that latency is known to induce, in which operators issue a discrete command,

suspend input, wait for the delayed consequence to become visible, and only then act again. The reduction in idle fraction indicates that the predictive display suppressed this adaptation, and did so to a degree that did not degrade over the delays we tested. The increase in reversals is less conclusive: it is consistent with continuous closed-loop correction, but since reversal rate also rises with prediction error (Sec. 4.2), part of it likely reflects corrections of inaccurate predictions.

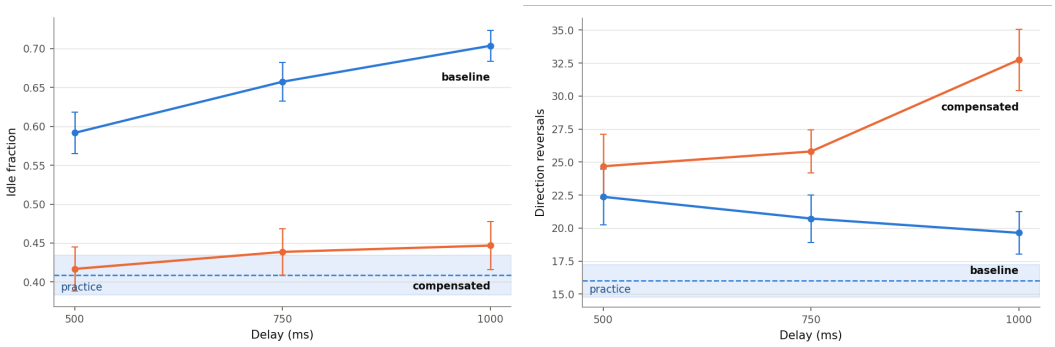


Figure 3: Control behaviour. **Left:** The idle rate was significantly higher in the baseline trials describing the known move-and-wait strategy under latency which the predictive display successfully suppressed. **Right:** The input reversal rate was higher in the compensated trials which might indicate more corrective behaviour due to visual artifacts.

**Prediction fidelity** Rollout fidelity degraded with delay, with mean LPIPS rising from 0.056 to 0.069 and 0.089 and mean SSIM falling from 0.885 to 0.867 and 0.844 at 500, 750 and 1000 ms; Fig. 6 in Appendix B.7 shows representative rollouts at each delay. Prediction error was associated with operator performance: within every delay level, trials with higher LPIPS took longer to complete (Spearman $\rho = 0.60, 0.37, 0.68$; all $p < .002$) and were more likely to reach the time cap ($\rho = 0.29, 0.28, 0.55$; all $p < .01$), with the same pattern for lower SSIM (Fig. 4). Higher LPIPS was also associated with a higher input reversal rate, though more weakly ($\rho = 0.22, 0.24, 0.31$; $p = .060, .042, .021$). Because these relationships held within delay levels, they are not an artifact of both quantities increasing with latency, although they may partly reflect trial difficulty, as states involving more contact are harder both to predict and to complete. Participants were sensitive to this degradation: at the condition level, displays with higher prediction error were rated as less authentic ($\rho = -0.49$, $p < .001$), and the association persisted within participants ($\rho = -0.43$, $p < .001$).

**Subjective workload and display trust** Participants rated the compensated display as markedly more responsive than delayed video alone ($\Delta = 20.1$, 95% CI $[6.9, 33.2]$; Fig. 5, Table 2), and this advantage held at every delay ($\Delta = 16.6$, 21.4 and 22.3 at 500, 750 and 1000 ms) without changing reliably across them. The predictive display therefore preserved a sense of immediate control under delay, which is the effect it was designed to produce. None of the workload subscales differed reliably between conditions when pooled across delays (mental demand $-0.5$, 95% CI $[-8.2, 7.3]$; performance $-4.4$, $[-14.2, 5.5]$; frustration $+6.1$, $[-2.4, 14.6]$; Appendix B.5), although frustration under compensation rose at the longest delay (72.5 at 1000 ms vs. 56.4 and 56.1 at the shorter delays). Rated display authenticity — the extent to which participants believed what they saw reflected the true state of the system — was substantially lower under compensation ($\Delta = -28.7$, 95% CI $[-41.0, -16.5]$). This is consistent with the association between prediction error and authenticity ratings reported in Sec. 4.2. One plausible contributor is that each grounding pass replaces the latent context, so the displayed image shifts abruptly when a correction lands, and these corrections grow with $k + G$ (Fig. 6). The cost of compensation thus appears in how trustworthy the display seems rather than in task performance, for which we found no evidence that the benefit differed across the delays tested.

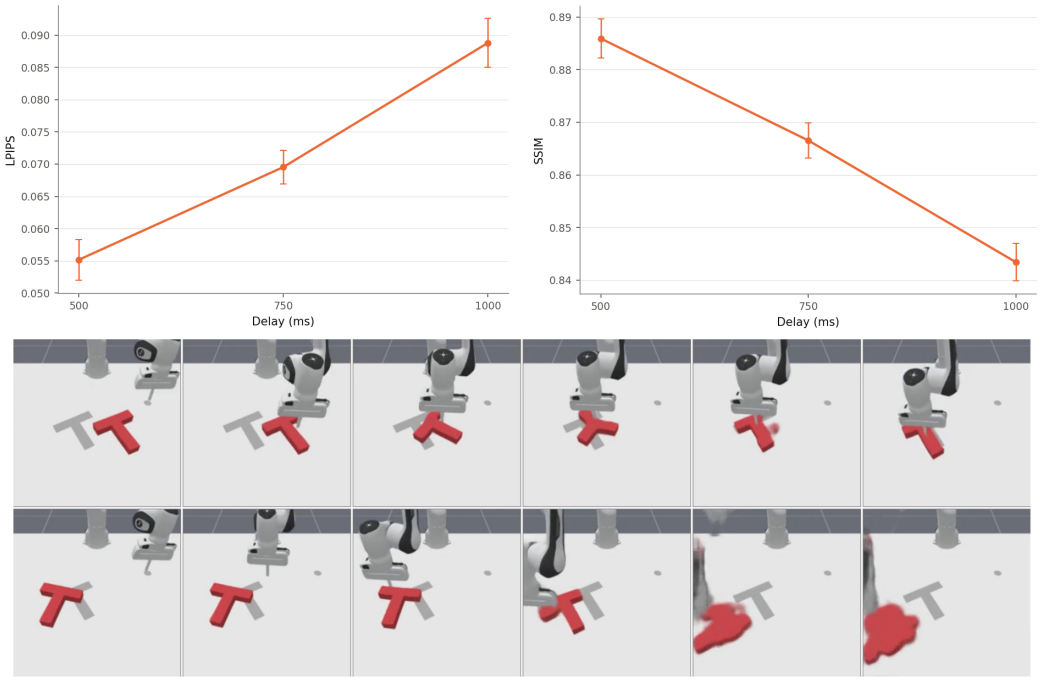


Figure 4: Prediction fidelity. **Top:** With increasing delay the visual fidelity declined due to longer rollouts and larger autoregressive drift. **Bottom:** Examples of visual artifacts.

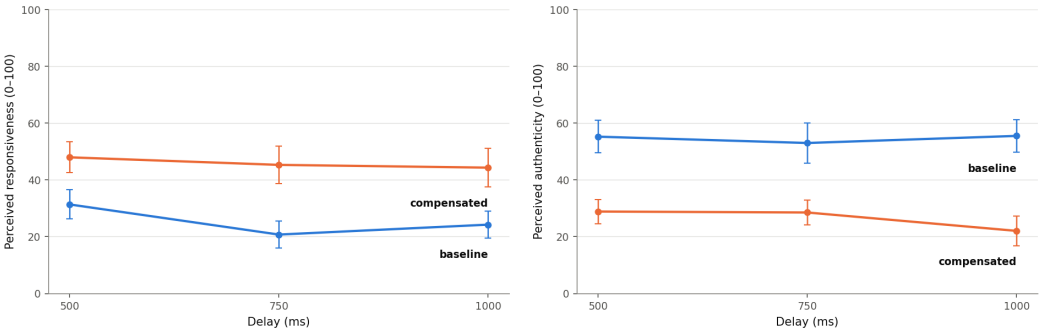


Figure 5: Subjective workload and display trust. **Left:** Participants reported better responsiveness under compensation. **Right:** Reported display authenticity was lower for the predictive display.

## 5 Conclusion

We presented a predictive display that uses an action-conditioned world model to compensate for network latency in teleoperation. A DINO-WM is rolled forward from the most recent received frame, with the computation split across a display thread that advances and decodes the estimate once per frame and an asynchronous grounding thread that re-anchors the shared latent context to ground truth. This decouples the per-frame cost from the delay and bounds the ungrounded prediction steps by $k + G$. In a within-subjects study with 22 participants on Push-T at round-trip delays of 500–1000 ms, compensation reduced completion time on successful trials by 23.5%, suppressed the move-and-wait strategy that latency induces, and preserved perceived responsiveness, none of which depended reliably on delay. Participants nevertheless rated the compensated display as less authentic, and these ratings tracked the degradation of prediction fidelity.

Table 2: Perceived responsiveness and display authenticity. Mean paired differences (compensated − delayed video) pooled across delays, with 95% CIs from the paired $t$ distribution. Higher scores indicate a more responsive or more authentic display. Workload subscales are reported in Appendix B.5.

| Item | $\Delta$ | 95% CI | $t$ | $p$ | $n$ |
|---|---|---|---|---|---|
| Responsiveness | 20.08 | [6.93, 33.22] | 3.18 | .005 | 22 |
| Authenticity | -28.71 | [-40.95, -16.47] | -4.88 | $< .001$ | 22 |

### 5.1 Limitations

**No delay on actions** We placed the entire round-trip delay on the video stream. In a real system part of it lies on the command channel, so the action issued at $t$ executes at $t + k_{\uparrow}$. The display would then have to predict ahead to the time a command takes effect, and the remote side would execute commands the operator issued from a prediction rather than an observation. Whether our results hold under such a split delay, and how the remote side should behave when its state diverges from the one the operator saw, remains open.

**Simple task** Push-T is a planar, rigid-body task with one manipulated object in simulation, and we further restricted the action space to constant-magnitude 2D directions. Real manipulation involves 6-DoF motion, multiple and possibly deformable objects, occlusions, and real camera footage, all of which make prediction substantially harder. Our results establish feasibility of the design rather than effectiveness in realistic settings.

**Online training** Training data came from a task policy rolled out with injected noise in parallel simulated environments, which presupposes a simulator and a policy that solves the task. Neither is generally available for real teleoperation, and training from logged operation instead reintroduces the coverage problem our scheme avoids: the off-policy interactions that operator commands produce at test time are scarce in such data.

**Variable delay** The procedure does not depend on a constant delay, but we evaluated only constant ones. Jitter and packet loss lengthen individual grounding passes and leave the reference frame older than expected; the effect on fidelity, on the frequency of visible corrections, and on operator performance is untested.

**Reliability** The predictor is deterministic and the display gives no indication of how reliable the current estimate is. The authenticity ratings show participants noticed wrong predictions, but a plausible wrong one can pass unnoticed and be acted on before ground truth arrives. Where actions are costly or irreversible, the display would need to communicate uncertainty and define its behaviour when prediction and ground truth diverge.

### Use of generative AI

In this work, we used generative AI tools for code generation and for writing assistance. In the user study, an LLM-based assistant was used to implement (i) the graphical user interface and trial-control harness that presented trials and logged operator input, (ii) the script that generated the counterbalanced trial schedule, including the Williams design, the within-block condition alternation and the per-trial initial-pose seeds, and (iii) the scripts that computed the reported statistics from the logged trial data. Prose in this paper was drafted by an LLM from bullet points, outlines and numerical results supplied by the authors. We did not use generative AI tools for research ideation, for the design or derivation of the method, for literature search or citation generation, or for generating figures.

We have reviewed all AI-assisted work. The study interface and schedule generator were read in full before data collection began. The generated schedule was verified against its design properties independently of the code that produced it. Every quantitative claim in the text and tables was checked against the analysis output, and no statement was retained on the basis of a model's assertion

alone; several discrepancies introduced during drafting were identified and corrected in this way. The authors take responsibility for the final content of this work, including all text, claims and artifacts produced with the aid of generative AI.

### ETHICS STATEMENT

This work includes a user study with 22 adult volunteers performing a simulated manipulation task with a game controller. Participation was voluntary, participants gave written informed consent after being informed of the task, the data recorded and their right to withdraw at any time, and no remuneration was offered beyond refreshments. The study involved no invasive, medical or diagnostic procedures and no vulnerable populations. No personally identifying information was recorded and no identifier-to-individual key was created, so the dataset is anonymous. Under the ethics commission statute of our institution, consultation is directed at research processing personal or personally identifiable data; this study processes none, and no application was required.

We release the anonymised trial-level study data, comprising task performance, control-input and prediction-fidelity measures together with coarse demographic bands. No raw video of participants was recorded at any point.

Our stated motivation is latency compensation for teleoperation, and we note a finding with direct bearing on that application. Participants performed the task faster under compensation while rating the display as substantially less authentic, and reported frustration increased at the longest delay, where prediction fidelity was lowest. A predictive display can therefore improve measured performance while degrading the operator's warranted confidence in what they are seeing. We also observe that operator performance degrades systematically with prediction error, which implies that a deployed system would require online fidelity monitoring and a means of signalling low-confidence predictions to the operator.

### REPRODUCIBILITY STATEMENT

The method is specified in full in Sec. 4: Algorithm 1 gives the two-thread display and grounding procedure, and the feasibility conditions and effective horizon $k+G$ are derived in the accompanying text, with the measured predictor and decoder latencies on our hardware. World-model training is described in Appendix A, including the PPO policy and its hyperparameters, the noise ladder used to generate rollouts, and the DINO-WM training settings; architecture and hyperparameters otherwise follow Zhou et al. (2024). The task is the unmodified ManiSkill PushT-v1 environment with the action-space restriction stated in Sec. 4 and the success criterion given in Appendix B.2.

The user study is documented in Appendix B. This covers the counterbalancing scheme and the rule assigning participants to schedule rows (Appendix B.4), the questionnaire items, anchors and scoring polarity (Appendix B.5), the definition of every reported metric (Appendix B.6), the time base and the treatment of censored trials (Appendix B.8), the pre-specified rule for handling simulator faults (Appendix B.9), and the statistical procedures with the model specifications and assumption checks (Appendix B.10).

We release the anonymised trial-level study data as supplementary material, one row per trial with condition, delay, schedule position, seed, and all performance, control-input, prediction-fidelity and questionnaire measures. Every number reported in Sec. 4 and Appendix B can be recomputed from this file using the procedures in Appendix B.10. We intend to release the implementation of the predictive display, the training pipeline and the study harness publicly; they are omitted from this submission because they are not yet in a state we can responsibly distribute, and we did not want to release code we had not finished reviewing.

## REFERENCES

Robert J Anderson and Mark W Spong. Bilateral control of teleoperators with time delay. In *Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics*, volume 1, pp. 131–138. IEEE, 1988.

Mads Antonsen, Siwen Liu, Xiao Xu, Eckehard Steinbach, Francesco Chinello, and Qi Zhang. Digital twin-empowered model-mediated teleoperation using multimodality data with signed distance fields. In *2024 IEEE Haptics Symposium (HAPTICS)*, pp. 353–359. IEEE, 2024.

Antal K Bejczy and Won S Kim. Predictive displays and shared compliance control for time-delayed telemanipulation. In *EEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications*, pp. 407–412. IEEE, 1990.

Andre Coelho, Harsimran Singh, Tin Muskardin, Ribin Balachandran, and Konstantin Kondak. Smoother position-drift compensation for time domain passivity approach based teleoperation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5525–5532. IEEE, 2018.

Etched Decart, Quinn McIntyre, Spruce Campbell, Xinlei Chen, and Robert Wachen. Oasis: A universe in a transformer. *URL: https://oasis-model. github. io*, 2(3):6, 2024.

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.

William R Ferrell. Remote manipulation with transmission delay. *IEEE Transactions on Human Factors in Electronics*, (1):24–32, 1965.

William R Ferrell and Thomas B Sheridan. Supervisory control of remote manipulation. *IEEE spectrum*, 4(10):81–88, 2009.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 2786–2793. IEEE, 2017.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019.

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

Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models. *arXiv preprint arXiv:2509.24527*, 2025.

Blake Hannaford and Jee-Hwan Ryu. Time-domain passivity control of haptic interfaces. *IEEE transactions on Robotics and Automation*, 18(1):1–10, 2002.

Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, et al. Matrix-game 2.0: An open-source real-time and streaming interactive world model. *arXiv preprint arXiv:2508.13009*, 2025.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

Dae-Kwan Ko, Dong-Han Lee, and Soo-Chul Lim. Continuous image generation from low-update-rate images and physical sensors through a conditional gan for robot teleoperation. *IEEE Transactions on Industrial Informatics*, 17(3):1978–1986, 2020.

Kang-Won Lee, Dae-Kwan Ko, Yong-Jun Kim, Jee-Hwan Ryu, and Soo-Chul Lim. Latency-free driving scene prediction for on-road teledriving with future-image-generation. *IEEE Transactions on Intelligent Transportation Systems*, 25(11):16676–16686, 2024.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

Siwen Liu, Mads Antonsen, Xiao Xu, Zhi Jin, and Eckehard Steinbach. Quality-of-task perception-based optimization of time-delayed teleoperation using a hybrid passivity-digital twin control strategy. In *2025 IEEE World Haptics Conference (WHC)*, pp. 301–307. IEEE, 2025.

Mitchell JH Lum, Jacob Rosen, Hawkeye King, Diana CW Friedman, Thomas S Lendvay, Andrew S Wright, Mika N Sinanan, and Blake Hannaford. Teleoperation in surgical robotics–network latency effects on surgical performance. In *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 6860–6863. IEEE, 2009.

Probal Mitra and Günter Niemeyer. Model-mediated telemanipulation. *The International Journal of Robotics Research*, 27(2):253–262, 2008.

MD Moniruzzaman, Alexander Rassau, Douglas Chai, and Syed Mohammed Shamsul Islam. Towards real-time generation of delay-compensated video feeds for outdoor mobile robot teleoperation. volume 106, pp. 48. Springer, 2022a.

MD Moniruzzaman, Alexander Rassau, Douglas Chai, and Syed Mohammed Shamsul Islam. High latency unmanned ground vehicle teleoperation enhancement by presentation of estimated future through video transformation. *Journal of Intelligent & Robotic Systems*, 106(2):48, 2022b.

MD Moniruzzaman, Alexander Rassau, Douglas Chai, and Syed Mohammed Shamsul Islam. Long future frame prediction using optical flow-informed deep neural networks for enhancement of robotic teleoperation in high latency environments. *Journal of Field Robotics*, 40(2):393–425, 2023.

Günter Niemeyer and J-JE Slotine. Stable adaptive teleoperation. *IEEE Journal of oceanic engineering*, 16(1):152–162, 2002.

Günter Niemeyer and Jean-Jacques E Slotine. Telemanipulation with time delays. *The International Journal of Robotics Research*, 23(9):873–890, 2004.

Florian Richter, Yifei Zhang, Yuheng Zhi, Ryan K Orosco, and Michael C Yip. Augmented reality predictive displays to help mitigate the effects of delayed telesurgery. In *2019 international conference on robotics and automation (ICRA)*, pp. 444–450. IEEE, 2019.

Jacob Rosen and Blake Hannaford. Doc at a distance. *IEEE spectrum*, 43(10):34–39, 2006.

Gaurav Sharma, Jeff Calder, and Rajesh Rajamani. Predictive display for teleoperation based on vector fields using lidar-camera fusion. *International Journal of Computer Vision*, 133(12):8315–8331, 2025.

Yishuang Sun, Fanle Meng, Dong Yang, Mengchen Xiong, and Xiao Xu. Low-cost modeling and haptic rendering for membrane-like deformable objects in model-mediated teleoperation. *IEEE Access*, 12:141198–141210, 2024.

Yixuan Wang, Rhythm Syed, Fangyu Wu, Mengchao Zhang, Aykut Onol, Jose Barreiros, Hooshang Nayyeri, Tony Dear, Huan Zhang, and Yunzhu Li. Interactive world simulator for robot policy training and evaluation. *arXiv preprint arXiv:2603.08546*, 2026.

Bert Willaert, Hendrik Van Brussel, and Günter Niemeyer. Stability of model-mediated teleoperation: Discussion and experiments. In *International conference on human haptic sensing and touch enabled computer applications*, pp. 625–636. Springer, 2012.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Kang-Il Yoon, Dae-Kwan Ko, and Soo-Chul Lim. Real-time video prediction using gans with guidance information for time-delayed robot teleoperation. *International Journal of Control, Automation and Systems*, 21(7):2387–2397, 2023.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

# A TRAINING

Training data for the DINO-WM was generated by rolling out a PPO policy trained with ManiSkill's PushT-v1 environment. The policy was trained in $1024$ parallel environments for $5 \times 10^7$ environment steps, with rollout and episode length $200$, a batch of $204{,}800$ steps per iteration, $8$ update epochs over 32 minibatches, learning rate $3 \times 10^{-4}$ with Adam ($\epsilon_o = 10^{-5}$), discount $\gamma = 0.99$, GAE $\lambda = 0.9$, PPO clip $\epsilon_{clip} = 0.2$, zero entropy bonus, no learning-rate annealing, and dense normalized reward (final success $\approx 0.875$).

The policy acts in the same two-degree-of-freedom teleoperation action space used at deployment—a unit-circle fixed-magnitude $[dx, dy]$ command of $0.035$m per step through a `pd_ee_delta_pose` controller that keeps the push height ($0.015$m) and upright orientation fixed —and conditions an MLP on a single-step compact proprioception (end-effector (EE) position and velocity and T-Block goal pose); exploring starts are encouraged via an initial EE displacement of at least $0.12$m.

Data for training the DINO-WM was then generated by rolling out this frozen policy in $64$ parallel environments with a per-environment noise ladder: environment $i$ replaces the policy action with a random-direction action or an idle action with probability $p_i = \frac{i}{N-1}$, so the batch spans pure-policy (env $0$) to pure-noise behaviour. Camera renders of size $224 \times 224$ were normalized to $[-1, 1]$ and proprioception contains the EE position and velocity. The goal pose for the T-Block as well as the starting position of the EE were fixed. The DINO-WM was trained for $80000$ steps with batch size 32 on a single GPU with context length $C = 3$. Despite the data pipeline, our training, including the choice of hyperparameters and architecture is consistent with the original paper Zhou et al. (2024).

# B USER STUDY DETAILS

This appendix documents the design, procedure, instrumentation and analysis of the user study reported in Sec. 4, together with the measures and subscales omitted from the main text.

## B.1 PARTICIPANTS

We recruited 22 participants from our lab. Participation was voluntary and unremunerated beyond refreshments offered after the session. Participants were informed in writing about the task, the data recorded, and their right to stop at any time, and gave written informed consent. The study involved no invasive, medical or diagnostic procedures and recorded no personally identifying information: participants were assigned a sequential pseudonymous identifier at enrolment, and no key linking identifiers to individuals was created or retained, so the resulting dataset is anonymous.

## B.2 APPARATUS AND TASK

Participants sat at a display and controlled the end effector with the left analog stick of a PlayStation controller. Stick deflection was thresholded at 0.15 of full deflection and mapped to a command of fixed unit magnitude in the stick's direction; deflections below threshold produced the idle action. The nominal control and rendering rate was 20 Hz (Appendix B.8) — the measured per-run mean was 19.3 Hz (min 15.0, max 20.0).

The task was Push-T in ManiSkill 3.0.1, with end-effector orientation and $z$-translation held fixed as described in Sec. 4. A trial was scored as successful when the T-block pose entered a tolerance of 0.8 — the fraction of the 2D goal-T area covered by the pushed block (area-overlap intersection, evaluated every tick). Trials not meeting this criterion within 60s were terminated and recorded as timed out. Initial block and end-effector poses were fixed.

Table 3: Participant demographics and self-reported prior experience ($n = 22$).

| Attribute | Level | $n$ |
|---|---|---|
| Age band | 18–24 | 2 |
| | 25–34 | 20 |
| Gender | Female | 4 |
| | Male | 17 |
| | Prefer not to say | 1 |
| Handedness | Right | 19 |
| | Left | 3 |
| Gaming, h/week | $< 1$ | 12 |
| | 1–5 | 6 |
| | 5–10 | 4 |
| Gamepad, h/week | $< 1$ | 19 |
| | 1–5 | 2 |
| | 5–10 | 1 |

### B.3 Procedure

Sessions followed a fixed sequence: consent and briefing, demographic questionnaire, practice and three measured delay blocks.

Practice consisted of 10 undelayed trials without compensation, run until completion or the 60 s cap. No performance feedback was given beyond the on-screen task state. Participants were offered a short break at each delay block transition.

### B.4 Design and counterbalancing

The study used a $3 \times 2$ within-subjects design crossing round-trip delay ($k = 10, 15, 20$ simulation ticks, i.e. 500, 750 and 1000 ms at 20 Hz) with display condition (compensated vs. delayed video). Each participant completed 4 trials per cell, for 24 measured trials and 528 measured trials in total.

**Delay block order** Delay was held constant within a block of 8 trials. Block order followed a Williams design, a Latin square balanced for first-order carryover, so that every delay follows every other delay equally often across participants. Because the number of delay levels is odd, the balanced design requires $2 \times 3 = 6$ orders rather than 3 (Table 4).

Table 4: Williams design for three delay levels (ms). Each row is crossed with the two possible starting display conditions, giving a schedule cycle length of 12.

| Order | Block 1 | Block 2 | Block 3 |
|---|---|---|---|
| 1 | 500 | 1000 | 750 |
| 2 | 750 | 500 | 1000 |
| 3 | 1000 | 750 | 500 |
| 4 | 750 | 1000 | 500 |
| 5 | 1000 | 500 | 750 |
| 6 | 500 | 750 | 1000 |

**Display condition order** Within each block the two display conditions alternated in an ABBA pattern (`A B B A A B B A`). This equalises the mean serial position of the two conditions within a block, so a linear within-block trend such as fatigue or warm-up cannot favour either, and places the critical comparison adjacent in time at matched delay. The condition opening each block alternated across blocks within a participant and across participants, so that neither condition is confounded with block position.

**Assignment** Participant $i$ was assigned schedule row $i \bmod 12$, where the 12 rows cross the 6 delay orders with the 2 starting conditions. The complete schedule was generated and version-controlled before recruitment began, and the experiment software read each participant's trial sequence from it rather than generating order at runtime. With 22 participants the cycle is incomplete: rows 0–9 were assigned to two participants and rows 10–11 to one, leaving a residual order imbalance of one participant per row.

**Questionnaire placement** Subjective ratings were collected after two designated trials per block, one per display condition (132 rated trials in total). Each designated trial was the second trial of a same-condition run within the ABBA sequence (positions 5 and 7), so that the rated trial was preceded by a trial of its own condition rather than of the other one. This attenuates the lag-1 contrast effect that immediate alternation would otherwise induce. Which condition was rated first flipped with the block's starting condition and was therefore counterbalanced across participants.

### B.5 Questionnaire instrument

**Instrument** We administered three subscales of the raw (unweighted) NASA-TLX — mental demand, performance and frustration — together with two custom items on perceived responsiveness and display authenticity. Physical demand and temporal demand were omitted: physical demand has no meaningful variance in a thumbstick task, and temporal demand is ambiguous in a study whose manipulation is itself temporal, since participants cannot distinguish experimenter-imposed time pressure from the latency under test. Effort was also omitted. The weighting procedure of the full NASA-TLX was not used.

**Response format** Each item was presented as a horizontal line divided into 20 intervals by 21 tick marks, with verbal anchors at the ends only and no numeric or midpoint labels. Responses were recorded as the tick index and rescaled to 0–100 in steps of 5. The control was presented unset and required a deliberate interaction.

**Item wording and polarity** Table 5 gives the exact wording and anchors. **Polarity is not uniform across items.** Mental demand and frustration are scored so that higher is worse. Performance, responsiveness and authenticity are scored so that higher is better; the performance item was presented with reversed anchors relative to the standard NASA-TLX form, in which "Perfect" occupies the low end. A negative paired difference on performance therefore indicates *worse* perceived performance under compensation.

Table 5: Questionnaire items as presented. Polarity indicates the direction of higher scores.

| Item | Prompt | Anchors | Polarity |
|---|---|---|---|
| Mental demand | How mentally demanding was the task? | Very low — Very high | higher = worse |
| Performance | How successful were you in accomplishing what you were asked to do? | Failure — Perfect | higher = better |
| Frustration | How insecure, discouraged, irritated, stressed and annoyed were you? | Very low — Very high | higher = worse |
| Responsiveness | The system responded immediately to my inputs. | Strongly disagree — Strongly agree | higher = better |
| Authenticity | I trusted what I saw on screen to reflect the real robot state. | Strongly disagree — Strongly agree | higher = better |

**Full results** Table 6 reports all five items pooled across delays; Table 7 gives the breakdown by delay. The three workload subscales did not differ reliably between conditions, but the confidence intervals span roughly 16–20 points of a 100-point scale, so modest effects in either direction cannot be excluded. Frustration under compensation was comparable to baseline at the two shorter delays and higher at 1000 ms (72.5 vs. 56.8), consistent with the fidelity degradation reported in Sec. 4.2; the corresponding interaction was not tested formally given the number of rated trials per cell.

Table 6: All questionnaire items. Mean paired differences (compensated − delayed video) pooled across delays, 95% CIs from the paired $t$ distribution. Polarity is given in Table 5 and is not uniform across items.

| Item | $\Delta$ | 95% CI | $t(21)$ | $p$ | $n$ |
|---|---|---|---|---|---|
| Mental demand | $-0.45$ | $[-8.23,\ 7.32]$ | $-0.12$ | .904 | 22 |
| Performance | $-4.39$ | $[-14.24,\ 5.46]$ | $-0.93$ | .364 | 22 |
| Frustration | 6.14 | $[-2.36,\ 14.63]$ | 1.50 | .148 | 22 |
| Responsiveness | 20.08 | $[6.93,\ 33.22]$ | 3.18 | .005 | 22 |
| Authenticity | $-28.71$ | $[-40.95,\ -16.47]$ | $-4.88$ | $< .001$ | 22 |

Table 7: Questionnaire items by delay. Entries are condition means (delayed video / compensated).

| Item | 500 ms | 750 ms | 1000 ms |
|---|---|---|---|
| Mental demand | 57.3 / 57.3 | 61.4 / 56.6 | 64.3 / 67.7 |
| Performance | 60.7 / 63.9 | 46.6 / 50.7 | 53.4 / 33.0 |
| Frustration | 52.5 / 56.4 | 57.3 / 56.1 | 56.8 / 72.5 |
| Responsiveness | 31.4 / 48.0 | 21.4 / 42.7 | 22.0 / 44.3 |
| Authenticity | 55.2 / 28.9 | 52.5 / 27.5 | 56.8 / 22.0 |

### B.6 Metric definitions

All metrics were computed using simulated task time to account for deviations from the target 20Hz wall-clock rate (Appendix B.8).

**Idle fraction** Proportion of control steps in a trial at which the commanded action was the idle action.

**Reversals** Number of sign changes in the commanded direction. Reported as a rate, dividing by completion time, since the raw count is confounded with trial duration.

**Pushes** Number of distinct contact episodes between end effector and T-block.

**SSIM, LPIPS** Computed per displayed frame against the simulator's ground-truth frame at the corresponding time step, then averaged within the trial. LPIPS used the AlexNet backbone (lpips 0.1.4, CPU).

Table 8: Secondary behavioural metrics. Condition means and mean paired differences (compensated − delayed video) with 95% CIs, successful trials only, $n = 22$ participants.

| Metric | Delayed | Compensated | $\Delta$ | 95% CI | $p$ |
|---|---|---|---|---|---|
| Idle fraction | 0.657 | 0.450 | $-0.207$ | $[-0.253,\ -0.161]$ | $< .001$ |
| Reversal rate ($s^{-1}$) | 0.511 | 0.832 | $+0.321$ | $[0.241,\ 0.401]$ | $< .001$ |
| Pushes | 12.88 | 13.10 | $+0.22$ | $[-0.47,\ 0.90]$ | .518 |
| EE path length (m) | 3.169 | 4.169 | $+1.000$ | $[0.309,\ 1.692]$ | .007 |
| Block path length (m) | 0.524 | 0.511 | $-0.014$ | $[-0.071,\ 0.043]$ | .620 |

### B.7 Qualitative rollouts

### B.8 Time base and censoring

Completion time is reported in simulated task time, obtained by dividing the elapsed control steps of a trial by the nominal 20 Hz control rate. This is the appropriate unit for comparing operator performance across conditions, since it measures the task independently of the wall-clock cost of rendering and inference in our unoptimised research implementation. That cost is not symmetric across conditions: the baseline condition sustained the nominal rate (20.00, 20.00 and 19.97 Hz at 500, 750 and 1000 ms), whereas under compensation the realised rate fell to 19.15, 18.57 and

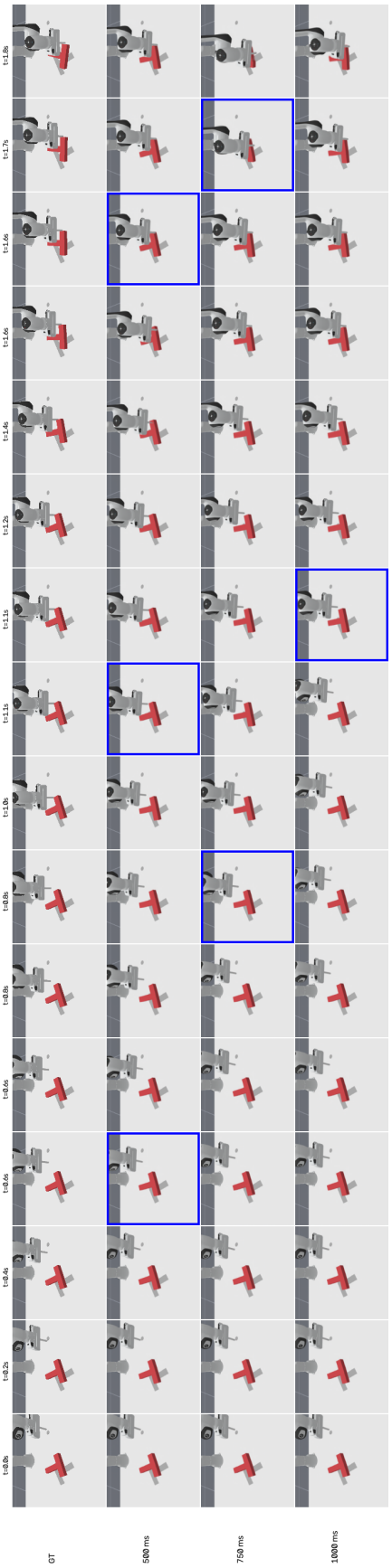


Figure 6: Frames produced by the display thread. Blue borders mark steps at which a grounding pass wrote back a corrected latent context. The visual discontinuity at these steps grows with the delay.

17.89 Hz, the deficit growing with the prediction horizon as expected from the $k + G$ analysis in Sec. 4.

One consequence must be stated explicitly. The 60 s trial cap was enforced on wall-clock time, so compensated trials reached the cap after fewer control steps than baseline trials: among capped trials, baseline runs terminated at 1200 steps, compensated runs at a mean of 1096 (range 902–1177). Censoring times expressed in simulated task time therefore vary across trials. The Cox model in Sec. 4 uses each trial's own censoring time rather than a common cap, which is the standard treatment of variable right-censoring and introduces no additional assumption. Descriptive comparisons on successful trials are unaffected, as they condition on completion. Reported in wall-clock time instead, the reduction on successful trials is 17.7% rather than 23.5%; we report the latter as the measure of operator performance and note the former as the figure a deployment on our current hardware would deliver.

### B.9 Data handling and exclusions

**Simulator faults** Trials terminated by unrecoverable simulator faults were handled under a rule specified before data collection: the trial is discarded and repeated with the next unused seed from a pre-committed per-participant pool of spare seeds, with condition and delay inherited from the discarded trial rather than chosen at analysis time. Of 528 measured trials, 498 were completed on the first attempt, 30 required a retrial.

**Distinguishing faults from task failures** Trials in which the participant failed to complete the task, including those reaching the time cap, are data and were retained. Only simulator faults were discarded.

**Censoring** 24.4% of measured trials reached the cap. The primary analysis treats these as right-censored at each trial's own censoring time (Sec. B.8). For the descriptive comparison on successful trials, capped trials are excluded; with capped trials included at their censoring value, the reduction in completion time is 18.2% rather than 23.5% ($t(21) = -4.15$, $p < .001$), which is conservative.

### B.10 Statistical analysis

Completion time was analysed with a Cox proportional hazards model with condition as a fixed effect, stratified by delay, and robust standard errors clustered by participant, which accounts for the repeated measurements without assuming a parametric form for between-participant variation. Stratification allows each delay level its own baseline hazard. This is necessary here: in an unstratified model with condition, delay and their interaction as fixed effects, the proportional hazards assumption was violated for the condition term (scaled Schoenfeld residuals, $\chi^2 = 19.10$, $p < .001$) and for the 1000 ms contrast ($\chi^2 = 5.79$, $p = .016$), reflecting the substantially different completion-rate profiles across delays. Under stratification no violation remains ($\chi^2 = 0.70$, $p = .403$). Because delay is a stratifying variable its main effect is not estimated as a hazard ratio and is reported descriptively in Sec. 4. The condition $\times$ delay interaction was tested by likelihood-ratio test against the stratified main-effects model.

Descriptive per-condition comparisons were computed on participant-level means, with completion time log-transformed before averaging because trial-level completion times are right-skewed and the effect of interest is multiplicative. Paired $t$ tests and 95% CIs from the paired $t$ distribution are reported for these comparisons. Trends across delay within a condition were tested with Friedman tests. Timeout rates were compared with a Wilcoxon signed-rank test on participant-level rates. Associations between prediction fidelity and behaviour used Spearman correlations computed within each delay level; the within-participant association for authenticity was obtained by centring both variables on their participant means before correlating, so that only variation within a participant contributes and between-participant differences in scale use are removed.

Error bars in figures showing within-subjects comparisons are Cousineau–Morey normalised confidence intervals; raw per-condition standard errors are not appropriate for judging these comparisons, as they include between-participant variance that the paired analysis removes.

Sample size was determined by an interim analysis after the first six participants. On the pilot data the compensated–baseline difference corresponded to $d_z = 0.848$ (completion time), which at $\alpha = .05$ (two-tailed) afforded 39.1% power at $n = 6$; a paired $t$-test power analysis (G*Power 3.1) indicated $N = 17$ for 90% power. We targeted 24, the smallest multiple of the 12-row schedule cycle (Sec. B.4) exceeding this requirement, and recruitment closed at 22. Because the pilot participants are included in the final sample, the effect-size estimate driving this decision is not independent of the reported result.

### B.11 Per-participant data

Table 9 reports each participant's condition means on successful trials. Two participants (P011, P012) were slower under compensation; both are retained in all analyses. Both nonetheless show the reduction in idle fraction and the increase in reversal rate seen throughout the sample.

Table 9: Per-participant condition means (delayed video / compensated), successful trials only.

| ID | Row | Completion time (s) | Idle fraction | Reversal rate ($s^{-1}$) |
|---|---|---|---|---|
| P001 | 0 | 35.6 / 25.5 | 0.59 / 0.36 | 0.60 / 0.86 |
| P002 | 1 | 26.7 / 22.7 | 0.70 / 0.50 | 0.43 / 0.97 |
| P003 | 2 | 35.2 / 27.9 | 0.72 / 0.43 | 0.40 / 0.86 |
| P004 | 3 | 30.8 / 19.8 | 0.63 / 0.43 | 0.61 / 1.08 |
| P005 | 4 | 32.5 / 24.6 | 0.64 / 0.46 | 0.72 / 0.82 |
| P006 | 5 | 32.7 / 29.5 | 0.35 / 0.16 | 0.83 / 1.37 |
| P007 | 6 | 27.0 / 14.4 | 0.58 / 0.41 | 0.53 / 0.71 |
| P008 | 7 | 27.6 / 16.8 | 0.62 / 0.46 | 0.62 / 1.00 |
| P009 | 8 | 34.5 / 25.1 | 0.81 / 0.70 | 0.28 / 0.33 |
| P010 | 9 | 29.6 / 23.3 | 0.68 / 0.45 | 0.48 / 0.82 |
| P011 | 10 | 32.9 / 40.4 | 0.55 / 0.42 | 0.49 / 0.70 |
| P012 | 11 | 37.5 / 39.5 | 0.73 / 0.31 | 0.44 / 0.93 |
| P013 | 0 | 41.8 / 30.2 | 0.70 / 0.65 | 0.56 / 0.74 |
| P014 | 1 | 27.1 / 20.1 | 0.64 / 0.47 | 0.51 / 0.91 |
| P015 | 2 | 43.7 / 36.6 | 0.71 / 0.39 | 0.41 / 0.75 |
| P016 | 3 | 29.3 / 26.2 | 0.78 / 0.55 | 0.46 / 0.72 |
| P017 | 4 | 27.8 / 23.6 | 0.76 / 0.53 | 0.42 / 0.74 |
| P018 | 5 | 42.6 / 31.7 | 0.62 / 0.53 | 0.46 / 0.53 |
| P019 | 6 | 31.7 / 24.8 | 0.55 / 0.20 | 0.59 / 1.28 |
| P020 | 7 | 39.3 / 19.6 | 0.74 / 0.51 | 0.53 / 0.75 |
| P021 | 8 | 42.0 / 31.2 | 0.65 / 0.66 | 0.44 / 0.48 |
| P022 | 9 | 32.3 / 23.9 | 0.70 / 0.31 | 0.45 / 0.97 |

### B.12 Limitations

- **Contrast effects in subjective ratings.** Conditions alternated within a block, so ratings were made with the alternative recently experienced. We placed rated trials after a same-condition trial to attenuate lag-1 contrast, but subjective differences should be read as an upper bound.
- **Correlational fidelity analysis.** The association between prediction error and operator performance may partly reflect a common cause in trial difficulty. Establishing direction would require degrading rollouts at matched task difficulty.
- **Incomplete schedule cycle.** With 22 participants and a 12-row cycle, ten rows were assigned twice and two once, leaving a residual order imbalance of one participant per row.