STORC: SPATIO-TEMPORAL OUTPUT-SPACE RESIDUAL CORRECTION FOR ROBOT VIDEO PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Robot video prediction models often lose accuracy in new scenes and tasks. As target frames are gradually revealed, they expose prediction errors and provide feedback for correcting future predictions, yet reliably and efficiently estimating future errors from limited history remains challenging. To address this challenge, we propose STORC, a spatio-temporal output-space residual correction framework requiring only one forward pass of the frozen backbone per prediction clip and no online gradient updates. STORC analytically estimates local relationships between prediction changes and residual changes at a regional scale, then extrapolates future residuals along the cached prediction trajectory, anchored at the latest observed residual. To control extrapolation risk, Causal Predictive Shrinkage (CPS) evaluates propagation against residual persistence on the latest completed transition held out from fitting and adaptively shrinks the propagation increment. To preserve fine spatial detail, STORC fuses regional and native-resolution corrections using the same shrunk propagation coefficients. Under a unified causal feedback protocol, STORC achieves an average MSE reduction of 34.8% relative to the strongest evaluated adaptation baseline in each of 45 backbone-split combinations, spanning five backbones and nine test splits on BridgeData, RoboNet, and BAIR. Core latency measurements show a geometric mean speedup of 3.91× over the evaluated adaptation baselines. Small-scale offline evaluation on self-collected laboratory videos demonstrates transfer without additional training or tuning.

1 INTRODUCTION

Video prediction infers scene evolution from observations, helping robots anticipate interaction outcomes and make decisions (Tian et al., 2025; Routray et al., 2026; Kim et al., 2026). However, source-trained predictors may lose accuracy in new scenes and tasks. As subsequent observations arrive, they reveal prediction errors and provide direct feedback for correcting future outputs. As illustrated in Figure 1(a), a short observed prefix provides few temporal observations but rich spatial error information. This feedback can improve predictions of unrevealed frames during deployment.

Reforecasting incorporates new observations by rerunning the backbone, incurring additional inference. Lightweight adaptation adjusts input or output modules using test-time feedback; some methods reuse forecasts but still optimize parameters online (Kim et al., 2025; Medeiros et al., 2025; Im & Kwon, 2026). Residual correction directly uses observed errors to improve future outputs through learned spatiotemporal mappings or temporal propagation without online gradient updates (Kim et al., 2022; Liu et al., 2026). Learned mappings require correction training, whereas analytic propagation primarily exploits temporal error structure. Applying these ideas to cached video also requires local estimation over dense outputs: spatially varying residuals must be estimated from few temporal transitions in a short prefix, limiting reliable independent pixelwise estimation.

Residual persistence provides direct, low-cost correction, but motion and appearance changes can alter future errors. Extrapolation from a short residual history can instead amplify estimation errors. Cached predictions offer additional information: even inaccurate trajectories encode estimates of local motion and appearance changes that may inform future residuals without further backbone inference. However, the relationship between prediction and residual changes can vary spatially and may not persist over time. Reliable correction therefore requires estimating these local relationships and assessing their propagation value from the revealed history.

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: Motivation and deployment trade-offs of causal error correction for cross-domain robot video prediction. (a) Rollout errors accumulate over time and are further amplified under scene and task shifts, while revealed observations provide dense spatial feedback. (b) STORC directly corrects future predictions without reforecasting and parameter updates. (c) Accuracy-efficiency comparisons characterize its deployment trade-off.

We propose **STORC**, a Spatio-Temporal Output-space Residual Correction framework for correcting cached video forecasts using observations revealed within each clip. As illustrated in Figure 1(b), STORC directly corrects future outputs from error feedback with one frozen-backbone forward pass per clip and no online gradient updates. **Local Analytic Residual Modeling (LARM)** pools regional statistics from limited history to analytically relate local prediction and residual changes, then extrapolates future residuals from the latest observed residual along the cached trajectory. Since fitting the history does not guarantee useful propagation, **Causal Predictive Shrinkage (CPS)** compares propagation with residual persistence on a held-out historical transition and adaptively shrinks the propagation increment. Regional aggregation stabilizes estimation but can smooth spatial details. STORC therefore fuses regional and native-resolution residual estimates using shared shrunk coefficients, without independently fitting pixelwise coefficients.

We evaluate quality and computational cost under a unified causal feedback protocol, with representative comparisons in Figure 1(c). The full evaluation covers five backbones and nine test splits on BridgeData (Ebert et al., 2021), RoboNet (Dasari et al., 2020), and BAIR (Ebert et al., 2017). Across the resulting 45 combinations, STORC reduces MSE by 34.8% on average relative to the strongest evaluated adaptation baseline in each setting. Core latency measurements yield a geometric mean speedup of 3.91× over the evaluated adaptation baselines. Component ablations distinguish basic residual correction gains from the incremental effects of propagation, shrinkage, and spatial branches. Small-scale offline evaluation on self-collected laboratory videos provides additional transfer evidence without further training or tuning.

Our main contributions are:

1. **Causal residual correction for cached video predictions.** STORC turns error feedback revealed within each clip into direct corrections of cached forecasts, allowing new observations to improve future outputs without additional backbone inference or online parameter updates.
2. **Prediction-driven local residual propagation and causal shrinkage.** LARM analytically estimates local propagation relationships from limited feedback, and CPS uses historical validation to regulate their propagation strength. The shrunk coefficients are shared by regional and native-resolution branches, combining reliability control with correction that balances estimation stability and spatial detail.
3. **Effectiveness and efficiency across backbones and distribution conditions.** We validate overall correction gains and computational efficiency across datasets, backbones, and distribution conditions, using component ablations to distinguish the contributions of basic residual correction and the added mechanisms. Offline evaluation on self-collected laboratory videos further examines transfer without additional training or tuning.

2 Related Work

2.1 Video Prediction for Robotics

Video prediction learns visual dynamics (Finn et al., 2016; Guen & Thome, 2020) through recurrent spatiotemporal modeling (Shi et al., 2015; Wang et al., 2017; 2019) or stochastic prediction of plausible futures (Babaeizadeh et al., 2017; Denton & Fergus, 2018; Lee et al., 2018). PredRNNv2 decouples recurrent memory states (Wang et al., 2022b); convolutional predictors such as SimVP, SimVPv2, TAU, and PFGNet emphasize efficiency (Gao et al., 2022; Tan et al., 2025; 2023; Cai et al., 2026), while PredFormer uses a pure Transformer architecture (Tang et al., 2026). These predictions also support control and planning (Finn & Levine, 2017; Ebert et al., 2018; Hafner et al., 2019). Predictive Inverse Dynamics Models condition action prediction on forecasted visual states (Tian et al., 2025), while Sparse Imagination reduces planning computation through sparse latent rollouts (Chun et al., 2026). Complementary to predictor and planner design, STORC corrects cached outputs using newly revealed observations.

2.2 Test-Time Adaptation for Forecasting

Test-time adaptation uses self-supervision (Sun et al., 2020; Han et al., 2026) or unsupervised objectives (Wang et al., 2020; 2022a; Niu et al., 2022), while forecasting additionally provides delayed target supervision (Kim et al., 2025; Im & Kwon, 2026). TAFAS and PETSA calibrate frozen forecasters through gated input/output mappings and low-rank modules, respectively (Kim et al., 2025; Medeiros et al., 2025). ST-TTC learns spectral-domain calibration (Chen & Liang, 2026), COSA uses context-aware gated residual correction (Im & Kwon, 2026), and δ-Adapter learns bounded input adjustments or output corrections (Liang et al., 2026). STORC instead estimates correction coefficients analytically from within-clip feedback, without online gradient updates.

2.3 Feedback-Based Residual Correction

ResCAL learns spatiotemporal residual mappings for traffic forecasting (Kim et al., 2022), while STEPS combines closed-form temporal smoothing with an offline-trained, forecast-conditioned decoder and cross-window error memory (Liu et al., 2026). Feedback World Model and FBFM incorporate feedback into latent predictions and flow-matching sampling, respectively (An et al., 2026; Li et al., 2026). Learned residual models require correction training, while latent and sampling feedback require access to model internals. With only cached RGB forecasts and a short observed prefix, spatially varying future errors must instead be estimated from few temporal samples. Persistence may miss local error changes, whereas extrapolation can amplify estimation errors. STORC addresses these difficulties by analytically relating local prediction increments to residual increments, while CPS controls propagation through held-out comparison with residual persistence.

3 Method

3.1 Problem Setup and Overview

Given an input sequence $X_{1:T}$, a frozen video predictor f_θ generates a future trajectory $\hat{Y}_{1:L} = f_\theta(X_{1:T})$, where each predicted frame has shape $H \times W \times 3$. The full trajectory is cached once and remains available throughout correction, while ground-truth frames are revealed causally over time. After the p-th target frame is revealed, only observations $O_{1:p}$ are available, whereas $O_{p+1:L}$ remain unknown. We define the revealed residuals as

$$\mathcal{R}_t = \mathcal{O}_t - \widehat{Y}_t, \qquad 1 \le t \le p. \tag{1}$$

Our goal is to estimate residuals for future frames $h > p$ using only the revealed prefix and the cached prediction trajectory. All residuals are defined relative to the original, uncorrected predictions, which remain fixed throughout the correction process.

As summarized in Figure 2, STORC consists of three stages. We first estimate a local relationship between prediction and residual changes from the revealed prefix. CPS then evaluates the reliability of this relationship on a held-out historical transition and shrinks unreliable propagation. Finally,

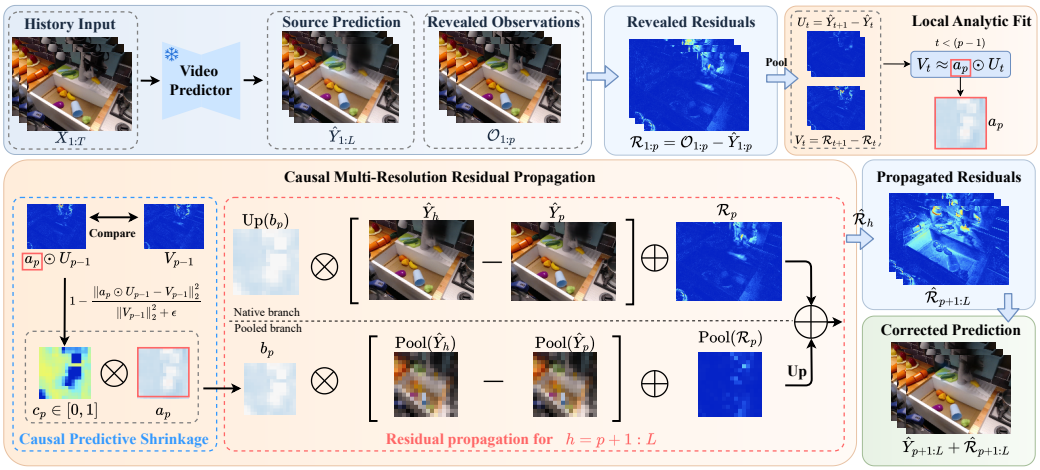


Figure 2: Overview of STORC after the p-th ground-truth frame is revealed. Only observations $\mathcal{O}_{1:p}$ are available; ground-truth frames beyond p remain unrevealed. Prediction and residual changes within the revealed prefix support local analytic estimation on a pooled grid. CPS compares propagation with residual persistence on the latest completed transition to obtain the shrinkage map c_p. Regional-scale and native-resolution branches share the resulting coefficients $b_p = c_p \odot a_p$ and extrapolate from the latest observed residual. The estimated residuals correct cached predictions for $h > p$, without additional backbone inference or gradient updates.

the resulting coefficients are applied at regional and native resolutions to estimate future residuals and correct the cached predictions.

3.2 Local Analytic Residual Modeling

To model residual evolution beyond the observed prefix, we relate residual increments to temporal changes in cached predictions. Since only a few transitions are available at test time, reliable estimation requires a low-complexity model that can aggregate local spatial evidence. We therefore adopt locally averaged scalar ridge regression (Hoerl & Kennard, 2000), which improves estimation stability through spatial aggregation and regularization while retaining a closed-form solution that requires no gradient updates.

We first project predictions and observed residuals onto a 16×16 grid using adaptive average pooling $\mathcal{P}$, and define adjacent temporal increments as

$$U_t = \mathcal{P}(\widehat{Y}_{t+1}) - \mathcal{P}(\widehat{Y}_t), \qquad V_t = \mathcal{P}(\mathcal{R}_{t+1}) - \mathcal{P}(\mathcal{R}_t), \qquad 1 \le t < p. \tag{2}$$

At each pooled location, we model the residual increment V_t as a scalar multiple of the prediction increment U_t, using one coefficient shared across RGB channels. This yields spatially varying propagation while keeping the number of estimated coefficients small under limited history.

To compensate for limited temporal evidence, we aggregate local statistics over a 3×3 neighborhood, assuming that nearby locations exhibit similar propagation behavior. At reveal step p, the observed prefix provides $p-1$ adjacent transitions. Let $\mathcal{N}_3$ denote neighborhood averaging with replicate padding. We compute

$$A_p = \mathcal{N}_3\left(\frac{1}{3(p-1)}\sum_{t=1}^{p-1}\|U_t\|_2^2\right), \quad B_p = \mathcal{N}_3\left(\frac{1}{3(p-1)}\sum_{t=1}^{p-1}U_t^\top V_t\right), \quad a_p = \frac{B_p}{A_p + \lambda_p}. \tag{3}$$

Here, A_p measures the local energy of prediction increments, while B_p measures their local alignment with residual increments. Inner products and norms are taken over RGB channels, and all divisions are pointwise on the pooled grid. We set $\lambda_p = \rho A_p + \epsilon$, with $\rho = 0.1$ and $\epsilon = 10^{-6}$. The term ρA_p controls the relative strength of regularization, while ϵ provides a denominator floor when prediction increments are small.

3.3 CAUSAL PREDICTIVE SHRINKAGE

Ridge regression can stabilize estimation from limited history, but it does not establish whether the estimated propagation relationship benefits subsequent predictions. In particular, when residuals change little, even a small extrapolation error may not improve upon residual persistence. CPS therefore uses residual persistence as a reference, assessing the benefit of extrapolation on the latest completed historical transition to adjust propagation strength.

CPS is activated when $p \geq 4$, ensuring at least two fitting transitions and one held-out transition for validation. First, a_p is fitted on transitions $t = 1, \ldots, p-2$ using the estimator in Section 3.2, with the latest completed transition (U_{p-1}, V_{p-1}) held out for validation. For this transition, residual extrapolation predicts an increment of $a_p \odot U_{p-1}$, whereas residual persistence predicts a zero increment. Their local errors are

$$\begin{aligned} E_p^{\text{prop}} &= \mathcal{N}_3\left(\frac{1}{3}\|a_p \odot U_{p-1} - V_{p-1}\|_2^2\right), \\ E_p^{\text{hold}} &= \mathcal{N}_3\left(\frac{1}{3}\|V_{p-1}\|_2^2\right). \end{aligned} \tag{4}$$

Here, $\odot$ denotes pointwise multiplication, with scalar coefficients broadcast over RGB channels. The held-out transition is excluded from the fit used to predict it, so this test measures out-of-sample predictive performance rather than in-sample fit. The entire procedure uses only the currently revealed history.

The shrinkage weight is defined by the relative error reduction of extrapolation over residual persistence:

$$c_p = \text{clip}_{[0,1]}\left(1 - \frac{E_p^{\text{prop}}}{E_p^{\text{hold}} + \epsilon}\right). \tag{5}$$

A larger relative benefit on the held-out transition retains more of the propagated increment, whereas a limited benefit reduces its influence. Setting $c_p = 0$ recovers residual persistence. Unlike a binary decision to enable or disable propagation, this continuous adjustment avoids turning a small historical advantage into a full extrapolation correction. Shrinkage acts only on the propagated increment because uncertainty concerns unobserved error changes, not the observed residual anchor.

Once c_p is determined, it is held fixed while a_p is refitted using all revealed transitions $t = 1, \ldots, p-1$. The effective coefficient for future propagation is then

$$b_p = c_p \odot a_p. \tag{6}$$

Thus, held-out validation controls extrapolation strength, while the full history is used to estimate the subsequent propagation relationship. When $p < 4$, we set $b_p = 0$ and use residual persistence.

3.4 DUAL-RESOLUTION CORRECTION AND CAUSAL EXECUTION

Spatial aggregation of regional-scale residuals and prediction changes reduces local fluctuations but also smooths fine-scale structures. Native-resolution information preserves these details, yet limited history makes independent pixelwise estimation of propagation relationships unsuitable. We therefore apply the same propagation coefficients at regional and native resolutions to balance estimation stability and spatial detail.

For a future frame $h > p$, both branches extrapolate from the latest observed residual using changes in cached predictions from p to h. Let $\mathcal{U}$ denote bilinear upsampling to the original resolution. The two branches estimate residuals as

$$\begin{aligned} \widehat{\mathcal{R}}_{p\to h}^{\text{reg}} &= \mathcal{P}(\mathcal{R}_p) + b_p \odot \left[\mathcal{P}(\widehat{Y}_h) - \mathcal{P}(\widehat{Y}_p)\right], \\ \widehat{\mathcal{R}}_{p\to h}^{\text{nat}} &= \mathcal{R}_p + \mathcal{U}(b_p) \odot (\widehat{Y}_h - \widehat{Y}_p). \end{aligned} \tag{7}$$

Both branches share b_p, requiring no additional coefficient fitting at the native resolution. The native branch preserves spatial detail through the original residual anchor and unpooled prediction changes, rather than by upsampling low-resolution residuals.

The regional estimate is upsampled and combined with the native-resolution estimate to obtain the final correction:

$$\begin{aligned} \widehat{\mathcal{R}}_{p\to h} &= 0.5\,\mathcal{U}(\widehat{\mathcal{R}}^{\mathrm{reg}}_{p\to h}) + 0.5\,\widehat{\mathcal{R}}^{\mathrm{nat}}_{p\to h}, \\ \widetilde{Y}_{p\to h} &= \mathrm{clip}_{[0,1]}\Big(\widehat{Y}_h + \widehat{\mathcal{R}}_{p\to h}\Big). \end{aligned} \tag{8}$$

Causal execution. Suppose ground-truth observations are revealed every d frames. At reveal step p, corrected outputs are committed only for frames $p+1$ through $\min(p+d, L)$, whose ground truth is not yet available. The first d frames use the original predictions clipped to $[0, 1]$. When new observations arrive, the correction is recomputed from the enlarged revealed history. Previously evaluated outputs are not revised, and corrected predictions are not recursively used for residual estimation. All within-clip state is cleared at the end of each clip and is not retained across clips.

4 Experiments

4.1 Experimental Setup

Datasets and evaluation. Based on PredBench (Wang et al., 2024), we evaluate $2 \to 10$ forecasting on BridgeData, RoboNet, and BAIR. BridgeData and RoboNet each include one standard and three shifted splits; BAIR contributes its standard test split, giving nine splits in total. Table 1 reports MSE and AlexNet-based (Krizhevsky et al., 2012) LPIPS (Zhang et al., 2018). The appendix also reports MAE, SSIM (Wang et al., 2004), PSNR, and I3D-based (Carreira & Zisserman, 2017) FVD (Unterthiner et al., 2018).

Backbones and baselines. We evaluate five frozen prediction backbones: PredRNNv2 (Wang et al., 2022b), TAU (Tan et al., 2023), SimVPv2 (Tan et al., 2025), PredFormer (Tang et al., 2026), and PFGNet (Cai et al., 2026). Together with the nine test splits, these yield 45 evaluation combinations. Table 1 compares STORC with uncorrected Source and five parameter-updating adaptation baselines: TAFAS (Kim et al., 2025), PETSA (Medeiros et al., 2025), ST-TTC (Chen & Liang, 2026), COSA (Im & Kwon, 2026), and δ-Adapter (Liang et al., 2026). Full details of the data splits, baselines, feedback protocols, and evaluation settings are provided in Appendix A.

4.2 Main Results

Table 1 shows that STORC achieves the lowest MSE and LPIPS in every evaluation combination. With the strongest adaptation baseline selected separately for each metric and combination, STORC reduces MSE by 23.1–47.3% and LPIPS by 27.6–49.5%. Mean MSE reductions range from 34.2% to 35.4% across the five backbones, with gains on both standard and shifted splits supporting consistency across architectures and evaluation conditions. Full results are provided in Appendix A.9. Table 2 further shows that STORC achieves the lowest MSE in all 18 combinations of TAU and PFGNet across nine splits under $4 \to 10$ forecasting. This evaluation uses length-specific checkpoints and 14-frame test windows, demonstrating applicability to a different input length rather than isolating the effect of additional context.

4.3 Ablation Studies

Table 3 evaluates the regional branch, LARM, and CPS on all four BridgeData splits using TAU and PFGNet. The regional branch lowers MSE in all settings, and LARM further improves all three reported metrics. CPS adds consistent gains on the six shifted settings but slightly worsens MSE and SSIM on the two Standard settings, likely because LARM is already reliable in-domain and the short-prefix reliability test over-shrinks useful propagation increments. Even so, the final STORC configuration lowers MSE by 11.5–18.2% over Native Persistence.

4.4 Accuracy-Efficiency Trade-off

Figure 3 compares prediction quality and core latency on BridgeData Standard using TAU and PFGNet. STORC achieves lower MSE and LPIPS than Source and δ-Adapter on both backbones,

Table 1: Prediction accuracy comparison under $2 \rightarrow 10$ forecasting. Best and second-best results are bold and underlined, respectively.

Backbone		MSE ↓							LPIPS ↓						
	Split	Source	TAFAS	PETSA	ST-TTC	COSA	δ-Adapter	STORC	Source	TAFAS	PETSA	ST-TTC	COSA	δ-Adapter	STORC
TAU	B-Std.	479.1	475.4	484.3	479.6	479.4	468.4	**356.3**	.1412	.1397	.1427	.1412	.1413	.1375	**.0925**
	B-Task	618.0	587.6	523.9	611.3	611.1	576.1	**290.5**	.1161	.1161	.1151	.1161	.1161	.1137	**.0720**
	B-Scene	1325.2	1313.4	1181.5	1320.4	1286.9	1229.6	**627.0**	.1655	.1655	.1548	.1653	.1645	.1594	**.0992**
	B-Both	1256.7	1196.7	1076.2	1240.3	1183.1	1121.0	**720.2**	.1834	.1793	.1642	.1830	.1800	.1759	**.1163**
	R-Std.	58.3	57.7	57.7	58.0	57.6	58.1	**41.9**	.0594	.0591	.0593	.0592	.0593	.0592	**.0382**
	R-Dyn.	139.8	139.1	139.4	139.3	139.0	138.8	**77.5**	.1033	.1031	.1017	.1032	.1031	.1018	**.0613**
	R-Scene	101.8	100.9	100.5	101.4	100.8	100.5	**77.1**	.1093	.1095	.1093	.1092	.1096	.1063	**.0686**
	R-Joint	319.8	317.9	318.9	319.6	317.5	316.6	**195.3**	.1860	.1867	.1857	.1859	.1865	.1875	**.1162**
	BAIR	287.1	286.9	289.0	286.9	287.0	286.9	**203.3**	.2334	.2333	.2334	.2333	.2334	.2331	**.1177**
PFGNet	B-Std.	509.4	507.4	516.4	509.9	509.9	502.3	**371.1**	.1442	.1430	.1465	.1444	.1443	.1424	**.0938**
	B-Task	516.0	508.1	495.4	517.2	517.2	497.5	**277.0**	.1154	.1153	.1183	.1156	.1156	.1135	**.0723**
	B-Scene	1245.0	1227.3	1159.6	1240.2	1233.0	1173.1	**611.2**	.1741	.1734	.1567	.1740	.1738	.1673	**.1023**
	B-Both	1152.4	1121.2	1066.9	1143.6	1139.4	1072.4	**705.0**	.1854	.1816	.1717	.1853	.1850	.1782	**.1163**
	R-Std.	56.5	56.4	56.7	56.4	56.5	56.3	**41.2**	.0643	.0640	.0643	.0641	.0642	.0635	**.0400**
	R-Dyn.	147.3	143.2	142.3	145.6	142.7	144.8	**78.9**	.1105	.1103	.1099	.1103	.1102	.1090	**.0634**
	R-Scene	105.1	104.2	104.1	104.6	104.2	103.7	**77.3**	.1206	.1207	.1197	.1204	.1210	.1179	**.0726**
	R-Joint	321.6	319.2	318.8	320.9	316.2	318.9	**195.2**	.1932	.1926	.1950	.1929	.1932	.1931	**.1191**
	BAIR	287.2	287.0	288.5	287.0	287.1	287.0	**203.4**	.2326	.2324	.2318	.2322	.2322	.2330	**.1179**
SimVPv2	B-Std.	533.6	532.3	546.1	534.0	534.2	528.7	**382.3**	.1392	.1385	.1407	.1392	.1393	.1381	**.0918**
	B-Task	493.3	485.9	489.2	487.6	489.3	481.3	**269.0**	.1100	.1100	.1100	.1100	.1100	.1084	**.0699**
	B-Scene	1166.8	1160.4	1105.6	1166.5	1163.6	1122.5	**600.9**	.1549	.1545	.1503	.1547	.1545	.1518	**.0969**
	B-Both	1069.9	1057.8	1041.3	1064.9	1060.7	1016.6	**688.2**	.1734	.1724	.1660	.1730	.1726	.1688	**.1124**
	R-Std.	58.1	57.9	58.5	58.0	58.1	58.0	**42.1**	.0608	.0608	.0611	.0607	.0608	.0604	**.0387**
	R-Dyn.	141.4	140.6	140.9	140.7	140.9	140.2	**78.5**	.1014	.1015	.1054	.1013	.1010	.1002	**.0611**
	R-Scene	102.8	102.0	101.5	102.2	102.1	101.4	**77.6**	.1076	.1079	.1050	.1075	.1076	.1049	**.0685**
	R-Joint	322.3	321.8	323.1	322.0	321.8	319.6	**196.6**	.1797	.1800	.1840	.1796	.1795	.1825	**.1156**
	BAIR	287.5	287.0	290.5	287.2	287.3	287.3	**203.8**	.2311	.2307	.2270	.2311	.2309	.2300	**.1176**
PredRNNv2	B-Std.	519.6	519.1	526.5	520.0	520.0	514.7	**375.3**	.1376	.1373	.1388	.1374	.1376	.1361	**.0919**
	B-Task	488.2	484.7	481.0	485.2	486.0	476.1	**271.1**	.1079	.1084	.1088	.1079	.1079	.1076	**.0696**
	B-Scene	1294.8	1265.8	1132.8	1288.1	1274.5	1221.1	**623.5**	.1737	.1718	.1510	.1734	.1722	.1666	**.1033**
	B-Both	1203.8	1132.6	1065.5	1196.0	1166.9	1098.0	**722.2**	.1984	.1909	.1730	.1979	.1946	.1854	**.1215**
	R-Std.	54.1	54.1	54.4	54.1	54.1	54.1	**40.0**	.0561	.0561	.0562	.0561	.0561	.0562	**.0364**
	R-Dyn.	136.6	135.5	135.0	136.5	136.7	135.9	**76.5**	.1090	.1083	.1034	.1090	.1091	.1083	**.0635**
	R-Scene	98.5	98.4	98.0	98.3	98.5	97.9	**75.3**	.1104	.1103	.1084	.1104	.1105	.1086	**.0694**
	R-Joint	324.0	323.1	323.8	323.6	323.5	320.1	**194.5**	.1996	.1992	.1922	.1996	.1998	.2000	**.1198**
	BAIR	284.9	284.8	285.9	284.8	284.9	284.9	**202.4**	.2288	.2287	.2283	.2286	.2288	.2283	**.1155**
PredFormer	B-Std.	525.1	521.2	529.9	525.7	525.5	517.1	**376.2**	.1365	.1346	.1373	.1364	.1365	.1328	**.0923**
	B-Task	571.9	563.3	522.8	565.2	567.2	550.8	**293.2**	.1082	.1078	.1057	.1081	.1082	.1063	**.0703**
	B-Scene	1359.9	1302.4	1204.1	1356.1	1338.7	1260.5	**635.3**	.1627	.1597	.1478	.1624	.1622	.1576	**.1008**
	B-Both	1270.9	1201.0	1089.2	1261.3	1184.3	1139.1	**739.0**	.1830	.1727	.1615	.1826	.1757	.1734	**.1169**
	R-Std.	58.7	58.7	59.1	58.7	58.7	58.6	**42.1**	.0568	.0567	.0569	.0567	.0568	.0565	**.0374**
	R-Dyn.	143.2	142.9	143.5	142.7	143.1	141.5	**78.5**	.0987	.0986	.0977	.0986	.0986	.0979	**.0604**
	R-Scene	108.2	107.9	107.5	107.9	108.1	105.9	**78.1**	.1026	.1022	.1004	.1024	.1023	.1002	**.0673**
	R-Joint	340.9	337.4	336.3	340.6	334.9	329.9	**198.6**	.1759	.1758	.1785	.1758	.1772	.1787	**.1153**
	BAIR	293.8	293.8	293.0	293.7	293.9	293.6	**206.5**	.2311	.2310	.2303	.2309	.2310	.2311	**.1165**

B- and R- denote BridgeData and RoboNet; Std. denotes Standard. B-Task, B-Scene, and B-Both denote task, scene, and joint scene–task shifts. R-Dyn., R-Scene, and R-Joint denote dynamics, scene, and joint shifts. BAIR uses its standard test split. Ranks use unrounded scores; leading zeros are omitted from LPIPS values.

Table 2: MSE under $4 \rightarrow 10$ forecasting. Best and second-best results are bold and underlined.

	TAU							PFGNet						
Split	Source	TAFAS	PETSA	ST-TTC	COSA	δ-Adapter	STORC	Source	TAFAS	PETSA	ST-TTC	COSA	δ-Adapter	STORC
B-Std.	477.7	475.2	490.7	477.9	477.8	468.2	**370.0**	501.3	499.9	521.5	501.2	501.5	492.7	**382.4**
B-Task	462.3	452.9	455.9	462.1	461.9	440.6	**281.0**	451.6	443.4	438.1	451.3	451.0	434.7	**281.1**
B-Scene	1205.6	1135.3	1105.8	1204.5	1200.3	1125.9	**606.3**	1338.9	1224.3	1120.0	1335.6	1317.7	1213.3	**637.3**
B-Both	1168.7	1135.2	1093.3	1165.0	1156.5	1075.3	**733.7**	1181.0	1134.1	1119.6	1177.0	1171.4	1089.7	**741.2**
R-Std.	58.5	58.3	59.2	58.5	58.5	58.1	**42.0**	57.2	57.1	57.9	57.1	57.2	57.0	**41.4**
R-Dyn.	134.7	134.3	135.8	134.4	134.6	133.0	**76.5**	137.5	135.6	136.1	136.4	135.2	135.3	**77.4**
R-Scene	96.2	95.9	97.7	96.1	96.2	95.1	**77.2**	98.5	98.1	98.6	98.3	98.2	97.3	**77.7**
R-Joint	313.1	311.7	314.9	312.9	312.7	308.4	**198.6**	313.3	311.9	315.5	313.0	312.0	307.9	**198.7**
BAIR	269.6	268.7	273.6	269.0	269.0	269.1	**208.1**	267.7	267.6	275.0	267.5	267.6	267.5	**207.4**

Runs share the same 14-frame test windows but use separate campaigns; these windows differ from Table 1.

Table 3: Component ablations on the four BridgeData test splits under $2 \rightarrow 10$ forecasting. Lower MSE and higher SSIM and PSNR indicate better performance. PSNR is reported in dB.

Backbone		Standard			Task shift			Scene shift			Joint shift		
	Variant	MSE↓	SSIM↑	PSNR↑	MSE↓	SSIM↑	PSNR↑	MSE↓	SSIM↑	PSNR↑	MSE↓	SSIM↑	PSNR↑
TAU	Native Persistence	435.42	.8535	23.72	328.29	.8829	24.04	714.40	.8206	20.50	877.34	.7968	19.87
	+ Regional	363.88	.8558	24.21	301.94	.8821	24.16	655.76	.8199	20.36	764.28	.7997	20.12
	+ LARM	**351.97**	**.8578**	24.30	297.84	.8835	24.21	653.44	.8237	20.38	742.49	.8041	20.22
	+ CPS (**STORC**)	356.28	.8575	**24.30**	**290.50**	**.8840**	**24.31**	**627.03**	**.8247**	**20.57**	**720.21**	**.8054**	**20.38**
PFGNet	Native Persistence	447.69	.8519	23.57	319.77	.8830	24.17	693.81	.8219	20.65	835.64	.8018	20.13
	+ Regional	375.81	.8522	24.01	285.39	.8832	24.45	633.85	.8202	20.48	721.02	.8036	20.39
	+ LARM	**368.44**	**.8539**	24.07	282.25	.8843	24.49	630.15	.8237	20.50	712.17	.8068	20.44
	+ CPS (**STORC**)	371.06	.8538	**24.09**	**277.01**	**.8848**	**24.56**	**611.18**	**.8247**	**20.66**	**704.96**	**.8071**	**20.52**

Rows build cumulatively on preceding variants. Native Persistence holds the latest native-resolution residual. Regional adds regional residual persistence. LARM: Local Analytic Residual Modeling; CPS: Causal Predictive Shrinkage.

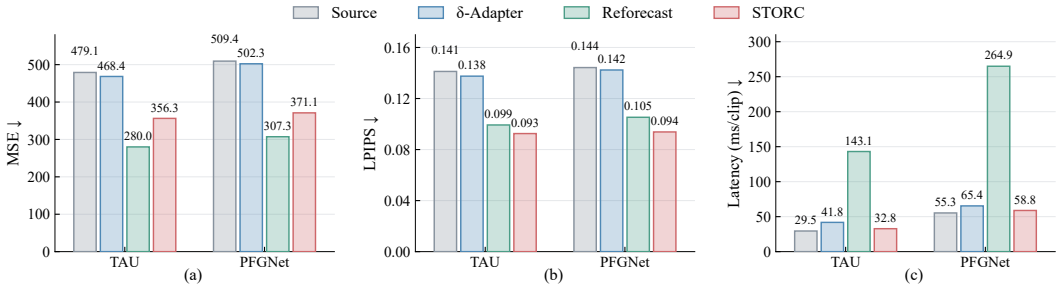


Figure 3: Accuracy–efficiency trade-offs on BridgeData Standard with TAU and PFGNet under $2 \rightarrow 10$ forecasting. (a) MSE, (b) LPIPS, and (c) core latency; lower is better for all three. Latency is measured on the first 512 clips with batch size 1 on an RTX 3090.

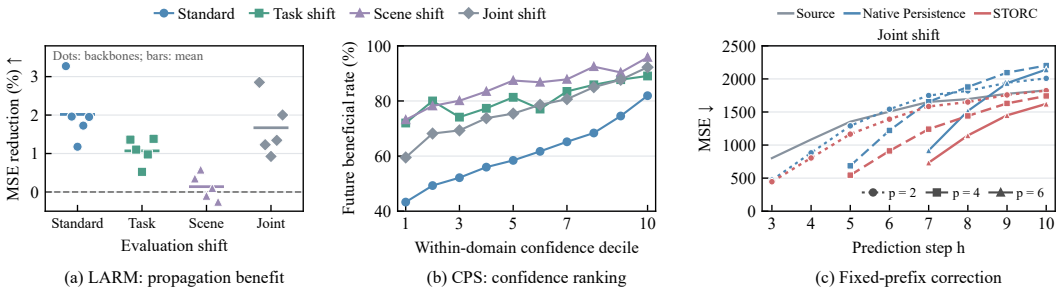


Figure 4: Mechanism and fixed-prefix analysis on BridgeData. (a) Relative MSE reduction of unshrunk propagation over residual persistence; positive values favor propagation. (b) Future propagation benefit by within-split confidence decile. (c) Per-step MSE on Joint shift using TAU, with feedback through $p = 2, 4, 6$ and no later observations.

with only 3.30 and 3.53 ms/clip of overhead over Source, respectively. Compared with Reforecast, STORC has higher MSE but lower LPIPS and runs $4.36\times$ and $4.50\times$ faster, using one backbone rollout instead of five. These results demonstrate the accuracy–efficiency trade-off of correcting cached predictions, with full timing results and protocols in Appendix A.5.

4.5 Mechanism and Sensitivity Analysis

We analyze whether residual propagation improves predictions, whether historical confidence predicts its benefit, and how correction quality evolves without further feedback on BridgeData.

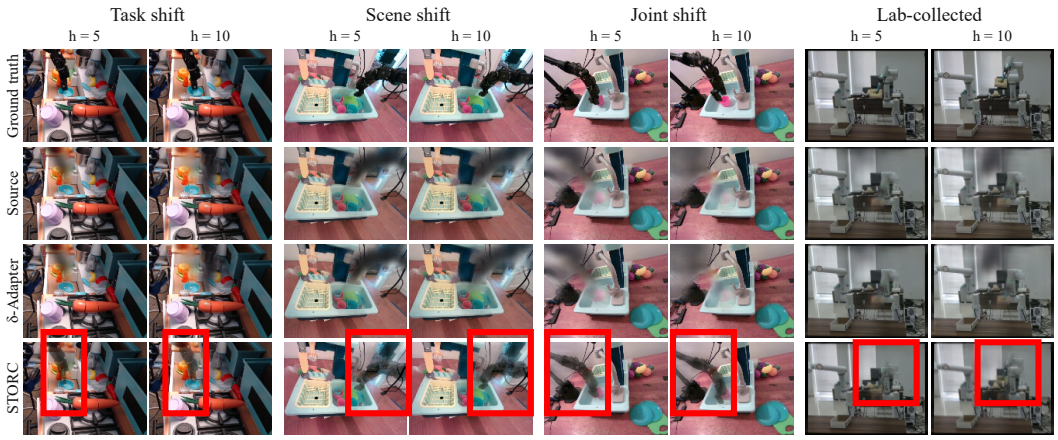


Figure 5: Qualitative comparisons using the frozen BridgeData-trained TAU under $2 \rightarrow 10$ forecasting. The four groups show BridgeData task, scene, and joint shifts, followed by a laboratory video (Lab-collected). Each group shows prediction steps $h = 5$ and $h = 10$, with ground truth, Source, δ-Adapter, and STORC arranged by row. STORC uses feedback revealed through steps $p = 4$ and $p = 8$, respectively. Red boxes highlight moving regions.

Propagation benefit and historical confidence. Figure 4(a) shows that unshrunk propagation reduces MSE over persistence in 18 of 20 backbone–split combinations with identical branches and gains, while two scene-shift settings degrade slightly. To assess whether historical feedback can identify reliable propagation, Figure 4(b) groups clip-averaged confidence at $p = 4, 6, 8$ into within-split deciles across all five backbones. Higher confidence generally corresponds to a greater fraction of subsequent two-frame blocks benefiting from propagation on all four splits. Thus, revealed history helps distinguish when propagation is more likely to improve future predictions, providing empirical support for confidence-guided shrinkage in CPS.

Correction after a fixed reveal. To assess how long a correction remains effective, we reveal targets through $p = 2, 4, 6$ and evaluate all remaining steps without further feedback. Figure 4(c) shows the per-step MSE of Source, Native Persistence, and STORC on Joint shift using TAU. STORC has lower MSE than both comparators at every evaluated step for all three prefixes, although errors still grow as predictions move farther beyond the last observation. Relative to Native Persistence, its mean MSE over the corresponding suffix decreases by 9.2%, 22.9%, and 24.0%, respectively. Complete results are provided in Appendix A.6.

4.6 Qualitative Results and Real-World Transfer

In Figure 5, STORC produces more distinct gripper and object boundaries under task shift and clearer robot-arm contours under scene and joint shifts. We further evaluate the frozen BridgeData-trained TAU on ten self-collected laboratory video clips without additional backbone training or STORC tuning. The Lab-collected group likewise shows clearer arm and object contours with STORC than with Source and δ-Adapter, although some blur and ghosting remain. This small-scale offline replay tests prediction transfer rather than closed-loop control. Detailed experimental settings and results are provided in Appendix A.8.

5 Conclusion

In this work, we proposed STORC, an output-space residual correction method for robot video prediction. It uses revealed errors to correct future outputs and assess propagation reliability without additional backbone inference or gradient updates. However, sparse feedback reduces perceptual quality, and some motion blur and ghosting remain. Our evaluation measures prediction quality rather than downstream control performance. Future work will address delayed or irregular feedback and test whether these improvements support more reliable planning and closed-loop robotic control.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing and language polishing, literature retrieval and discovery, research ideation and execution, and drafting sections of the manuscript. We did not use generative AI tools to generate datasets or prove mathematical claims. All AI-assisted content and outputs were reviewed by the authors. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Tuo An, Jindou Jia, Gen Li, Jingliang Li, Chuhao Zhou, Pengfei Liu, Bofan Lyu, Jiaqi Bai, Xinying Guo, Geng Li, et al. Feedback world model enables precise guidance of diffusion policy. *arXiv preprint arXiv:2605.15705*, 2026.

Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017.

Xinyong Cai, Changbin Sun, Yong Wang, Hongyu Yang, and Yuankai Wu. Pfgnet: A fully convolutional frequency-guided peripheral gating network for efficient spatiotemporal predictive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 38848–38858, June 2026.

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 4724–4733. IEEE, 2017.

Wei Chen and Yuxuan Liang. Learning with calibration: Exploring test-time computing of spatio-temporal forecasting. *Advances in Neural Information Processing Systems*, 38:155895–155929, 2026.

Junha Chun, Youngjoon Jeong, and Taesup Kim. Sparse imagination for efficient visual world model planning. In *International Conference on Learning Representations*, volume 2026, pp. 102160–102188, 2026.

Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura (eds.), *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 885–897. PMLR, 30 Oct–01 Nov 2020.

Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *International conference on machine learning*, pp. 1174–1183. PMLR, 2018.

Frederik Ebert, Chelsea Finn, Alex X. Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. In *1st Annual Conference on Robot Learning, CoRL 2017, Mountain View, California, USA, November 13-15, 2017, Proceedings*, volume 78 of *Proceedings of Machine Learning Research*, pp. 344–356. PMLR, 2017.

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.

Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021.

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.

Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. *Advances in neural information processing systems*, 29, 2016.

Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z Li. Simvp: Simpler yet better video prediction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3160–3170. IEEE, 2022.

Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 11471–11481. Computer Vision Foundation / IEEE, 2020.

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.

Dongchen Han, Yining Li, Tianyu Li, Zixuan Cao, Ziming Wang, Jun Song, Yu Cheng, Bo Zheng, and Gao Huang. ViT^3: Unlocking test-time training in vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 51–61, 2026.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 42(1):80–86, 2000.

Jeonghwan Im and Hyuk-Yoon Kwon. Cosa: Context-aware output-space adapter for test-time adaptation in time series forecasting. In *The Fourteenth International Conference on Learning Representations*, 2026.

Daejin Kim, Youngin Cho, Dongmin Kim, Cheonbok Park, and Jaegul Choo. Residual correction in real-time traffic forecasting. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 962–971, 2022.

HyunGi Kim, Siwon Kim, Jisoo Mok, and Sungroh Yoon. Battling the non-stationarity in time series forecasting via test-time adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 17868–17876, 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, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

Peize Li, Ruimeng Zhang, Ru Zhang, Cong Huang, Kai Chen, and Shanghang Zhang. Fbfm: A training-free asynchronous feedback mechanism for flow-matching in world-action models execution. *arXiv preprint arXiv:2607.29235*, 2026.

Daojun Liang, Qi Li, Yinglong Wang, Jing Chen, Hu Zhang, Xiaoxiao Cui, Qizheng Wang, and Shuo Li. The forecast after the forecast: A post-processing shift in time series. In *The Fourteenth International Conference on Learning Representations*, 2026.

Jiaqi Liu, Yifan Ouyang, Zhifei Song, Sim Kuan Goh, and Ashwaq Qasem. Steps: A temporal smooth error propagation solver on the manifolds for test-time adaptation in time series forecasting. *arXiv preprint arXiv:2605.08005*, 2026.

Heitor R Medeiros, Hossein Sharifi-Noghabi, Gabriel L Oliveira, and Saghar Irandoust. Accurate parameter-efficient test-time adaptation for time series forecasting. *arXiv preprint arXiv:2506.23424*, 2025.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pp. 16888–16905. PMLR, 2022.

Sandeep Routray, Hengkai Pan, Unnat Jain, Shikhar Bahl, and Deepak Pathak. ViPRA: Video prediction for robot actions. In *The Fourteenth International Conference on Learning Representations*, 2026.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 802–810, 2015.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pp. 9229–9248. PMLR, 2020.

Cheng Tan, Zhangyang Gao, Lirong Wu, Yongjie Xu, Jun Xia, Siyuan Li, and Stan Z Li. Temporal attention unit: Towards efficient spatiotemporal predictive learning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18770–18782. IEEE, 2023.

Cheng Tan, Zhangyang Gao, Siyuan Li, and Stan Z Li. Simvpv2: Towards simple yet powerful spatiotemporal predictive learning. *IEEE Transactions on Multimedia*, 27:5170–5184, 2025.

Yujin Tang, Lu Qi, Xiangtai Li, Chao Ma, and Ming-Hsuan Yang. Video prediction transformers without recurrence or convolution. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856.

Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 92033–92052, 2025.

Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7191–7201. IEEE, 2022a.

Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S. Yu. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 879–888, 2017.

Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3d LSTM: A model for video prediction and beyond. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Yunbo Wang, Haixu Wu, Jianjin Zhang, Zhifeng Gao, Jianmin Wang, Philip S Yu, and Mingsheng Long. Predrnn: A recurrent neural network for spatiotemporal predictive learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2208–2225, 2022b.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Zidong Wang, Zeyu Lu, Di Huang, Tong He, Xihui Liu, Wanli Ouyang, and Lei Bai. Predbench: Benchmarking spatio-temporal prediction across diverse disciplines. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LVIII*, volume 15116 of *Lecture Notes in Computer Science*, pp. 286–304. Springer, 2024.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

A Appendix

A.1 Datasets and Test Splits

We evaluate on BridgeData, RoboNet, and BAIR. BridgeData uses four splits: Standard, Original Scene–New Task, New Scene–Original Task, and New Scene–New Task, containing 3,970, 250, 318, and 645 test clips, respectively. Scene and task identities are defined by the dataset annotations relative to the source training set. The Standard split is partitioned at the camera-stream level; different camera views of the same physical trajectory may occur in training and testing. Our factorized RoboNet evaluation includes Standard, New Dynamics, New Scene, and Joint Shift, with 256 clips per split. These splits test changes in the motion policy, scene/camera configuration, or both, rather than treating dynamics shifts as task shifts. BAIR contributes its 256-clip standard test set. The main protocol predicts ten frames from two input frames, at 120×160 resolution for BridgeData and 64×64 for RoboNet and BAIR. The nine test splits and five backbones yield 45 evaluation combinations.

A.2 Backbones and Baseline Implementations

We use PredRNNv2, TAU, SimVPv2, PredFormer, and PFGNet, covering recurrent, convolutional, and Transformer-based predictors. Each backbone is trained on its source training split, and the checkpoint selected by validation MSE is frozen during evaluation.

Main-table baselines. Table 1 compares STORC with Source, which applies no correction, and five parameter-updating adaptation baselines: TAFAS, PETSA, ST-TTC, COSA, and δ-Adapter. These forecasting methods are implemented through video-compatible calibration interfaces. Our δ-Adapter comparator uses the additive output-correction variant, initialized as an identity mapping and adapted online without separate offline adapter training.

A.3 Causal Feedback Protocol

For STORC and Reforecast, ground-truth target frames are revealed every $d = 2$ frames. Updates occur after target frames $p = 2, 4, 6, 8$, and affect only the next unobserved two-frame block. The first two targets use the original predictions; all ten targets contribute to evaluation, and already scored outputs are never revised.

A.4 Metrics and Implementation

We report MSE, MAE, SSIM, PSNR, LPIPS, and FVD. All methods are evaluated on the same test clips using an identical evaluation pipeline, including the same revealed frames, scored prediction frames, metric implementations, and aggregation procedure. MSE and MAE are computed by summing errors over the spatial and RGB dimensions and then averaging over the ten target frames and test clips. LPIPS uses AlexNet features and is likewise averaged over target frames and clips. FVD is computed once for each backbone–split–method combination from all predicted and ground-truth test clips using I3D-400 features. Quality evaluation is performed on an RTX 4090. For STORC, backbone inference uses BF16 and residual estimation uses FP32. Unless otherwise specified, we use a 16×16 pooled grid, a 3×3 spatial averaging neighborhood, $\rho = 0.1$, and $\epsilon = 10^{-6}$. Residuals are always computed from the original cached backbone predictions, and clipping to $[0, 1]$ is applied only to the final corrected outputs. No hyperparameter is tuned separately for individual test splits.

A.5 Efficiency Measurement Protocol

Measurements use the same RTX 3090 GPU and the first 512 clips of BridgeData Standard in their original order. All methods use batch size 1, two input frames, ten predicted frames, and 120×160 resolution. Backbone inference uses BF16, while adapters retain their native internal precision and STORC estimates residuals in FP32. Core latency is measured from GPU-resident inputs to completed prediction, required adaptation or correction, and output clipping, with CUDA synchronization. It includes additional backbone rollouts, backpropagation, and optimizer steps when required by a method, but excludes data loading, decoding, preprocessing, host-to-device

transfer, model loading, warm-up, quality metrics, and external logging. These measurements are not end-to-end robot control latency.

Table 4 reports the mean and sample standard deviation of three complete 512-clip run means. In the same campaign, Source and STORC have identical measured peak allocated GPU memory: 320.6, 290.9, 300.4, 151.7, and 97.0 MiB for PredRNNv2, TAU, SimVPv2, PredFormer, and PFGNet, respectively. These values are the largest peaks across the three runs, not reserved memory or total device usage, and do not imply zero temporary allocation or runtime overhead. The final campaign's Source peak is 32 MiB below its historical value for each backbone, so cross-campaign memory differences cannot be attributed entirely to algorithm design.

For each backbone and adaptation baseline, speedup is the baseline mean latency divided by the STORC mean latency. The reported $3.91\times$ is the geometric mean of these 25 ratios, excluding Source and Reforecast. The Reforecast speedup range is computed separately across its five backbone-specific ratios.

Table 4: Core latency in ms/clip, reported as mean $\pm$ sample standard deviation of three 512-clip run means. These repetitions measure timing variability, not uncertainty across training seeds.

Method	PredRNNv2	TAU	SimVPv2	PredFormer	PFGNet
Source	63.56 ± 0.13	29.49 ± 0.29	27.71 ± 0.03	50.24 ± 0.19	55.30 ± 0.10
TAFAS	824.47 ± 0.07	415.70 ± 7.95	355.13 ± 3.61	695.02 ± 13.84	799.23 ± 11.33
PETSA	958.11 ± 0.70	470.22 ± 10.74	427.24 ± 6.10	811.87 ± 19.51	935.02 ± 20.16
ST-TTC	128.79 ± 0.47	59.14 ± 0.32	56.49 ± 0.20	101.34 ± 0.50	104.65 ± 1.00
COSA	115.93 ± 0.31	81.86 ± 0.76	79.74 ± 0.82	101.10 ± 1.18	104.86 ± 1.71
δ-Adapter	75.74 ± 1.24	41.81 ± 0.66	38.80 ± 0.33	61.13 ± 0.38	65.37 ± 1.72
Reforecast	317.38 ± 0.07	143.08 ± 2.88	137.21 ± 0.02	247.44 ± 2.90	264.94 ± 4.23
STORC	64.56 ± 0.02	32.79 ± 0.16	28.59 ± 0.03	54.88 ± 0.80	58.83 ± 0.61

A.6 Additional Feedback Analyses

Fixed-prefix evaluation. Figure 6 extends Figure 4(c) to TAU and PFGNet on all four BridgeData test splits under $2 \to 10$ forecasting. For each $p \in \{2, 4, 6\}$, only targets $1{:}p$ are revealed; corrections are computed once at p and evaluated over $h = p + 1, \ldots, 10$ without further feedback or backbone inference. Source uses the original cached predictions and is shared across prefixes. Native Persistence adds the latest native-resolution residual with unit gain. At $p = 2$, propagation and CPS are inactive, so STORC reduces to dual-resolution persistence. MSE is summed over spatial and RGB dimensions and averaged across clips at each prediction step; each reported relative reduction compares the two methods over the same suffix.

Sensitivity to feedback intervals. Figure 7 examines how feedback frequency affects perceptual quality on BridgeData Standard using TAU and PFGNet. The feedback interval varies over $d \in \{1, 2, 4, 8\}$, and LPIPS is averaged over all ten predicted frames and all 3,970 test clips. More frequent feedback improves LPIPS for STORC and Reforecast, while δ-Adapter changes little under the evaluated protocol. STORC achieves lower LPIPS than Reforecast at every interval on both backbones, with a smaller advantage as feedback becomes sparse. Thus, STORC benefits from timely residual feedback while retaining a single backbone forward pass per clip.

A.7 Pooling-Grid Sensitivity

We vary the pooled output resolution over 4×4, 8×8, 16×16, and 32×32 on all four BridgeData test splits using TAU and PFGNet under $2 \to 10$ forecasting. Unlike fixed-prefix evaluation, this experiment reveals feedback every two frames at $p = 2, 4, 6, 8$ and evaluates all ten predicted frames on each complete test split. Grid size denotes the pooled output resolution, not the pooling kernel size; the averaging neighborhood remains fixed in grid coordinates.

Table 5 shows that the MSE-optimal resolution varies across splits: 32×32 performs best on task and scene shifts for both backbones, whereas 16×16 performs best on joint shift. On Standard, TAU

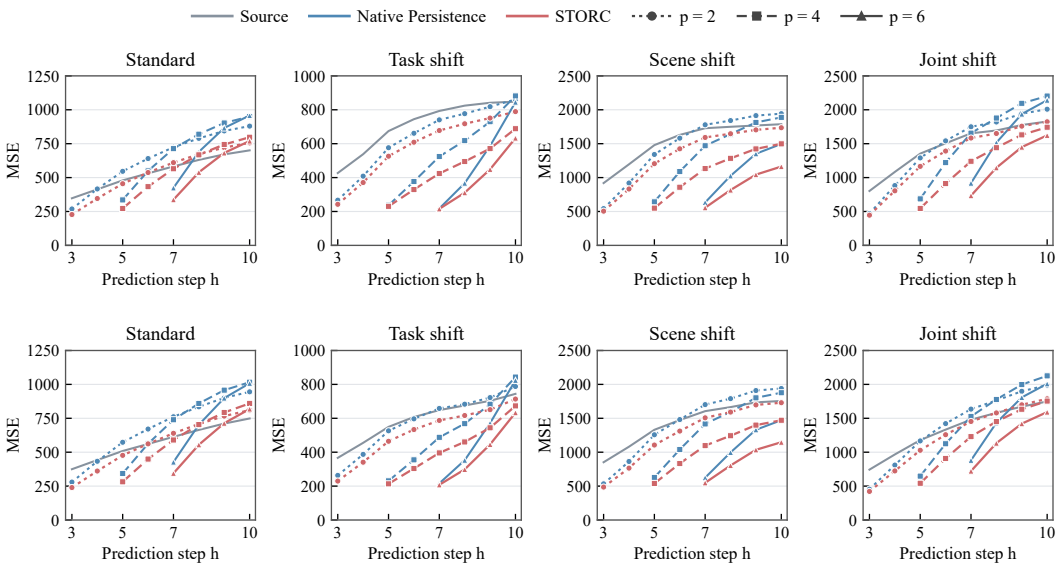


Figure 6: Fixed-prefix MSE on all four BridgeData test splits. The top and bottom rows use TAU and PFGNet, respectively. Gray, blue, and red indicate Source, Native Persistence, and STORC. Dotted lines with circles, dashed lines with squares, and solid lines with triangles indicate $p = 2$, $p = 4$, and $p = 6$, respectively. Each correction uses only targets $1{:}p$ and is evaluated at $h = p+1, \ldots, 10$, with no later feedback. Source is shared across prefixes and plotted once in each panel. The two rows use matched MSE scales within each split; lower is better.

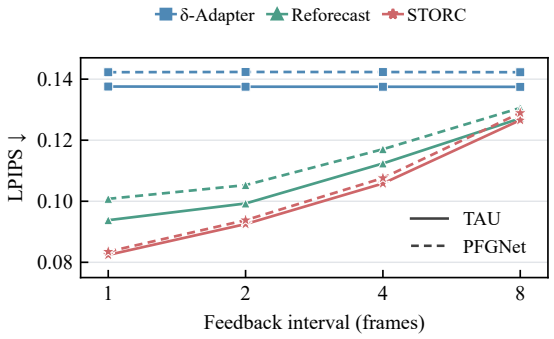


Figure 7: Feedback-interval sensitivity on BridgeData Standard under $2 \rightarrow 10$ forecasting. Colors indicate methods; solid and dashed lines indicate TAU and PFGNet, respectively. Lower LPIPS is better.

and PFGNet achieve their lowest MSE at 8×8 and 16×16, respectively. In contrast, 32×32 achieves the lowest LPIPS in all eight backbone–split combinations. These results characterize sensitivity to a shared grid size rather than selecting a separate configuration for each split.

We also measure correction-only latency on the same RTX 3090 used in Appendix A.5, using the identical first 512 Standard clips and batch size 1. After 32 warm-up clips per configuration, we time the complete four-reveal correction with CUDA synchronization, excluding backbone inference, data transfer, quality metrics, and diagnostic summaries. All grid sizes use the same generalized implementation, whose outputs match the quality evaluator exactly on 16 verification clips per backbone. Table 5 reports the mean and sample standard deviation of three complete timing runs, with shuffled grid order between runs. Latency remains within 3.83–3.91 ms/clip across the tested configurations, without a consistent monotonic increase across both backbones.

Table 5: Pooling-grid sensitivity on BridgeData using TAU and PFGNet under $2 \rightarrow 10$ forecasting. Lower MSE, LPIPS, and correction latency are better.

Backbone		Standard		Task shift		Scene shift		Joint shift		Correction
	Grid	MSE↓	LPIPS↓	MSE↓	LPIPS↓	MSE↓	LPIPS↓	MSE↓	LPIPS↓	(ms/clip)↓
TAU	4×4	361.21	.0936	350.73	.0761	728.83	.1037	772.46	.1218	3.833±0.057
	8×8	**355.41**	.0933	314.60	.0743	666.44	.1018	734.20	.1195	3.895±0.036
	16×16	356.28	.0925	290.50	.0720	627.03	.0992	**720.21**	.1163	3.861±0.072
	32×32	364.22	**.0881**	**282.89**	**.0668**	**618.16**	**.0929**	726.60	**.1097**	3.900±0.019
PFGNet	4×4	377.66	.0951	318.38	.0748	707.43	.1064	749.69	.1210	3.857±0.065
	8×8	371.18	.0947	295.22	.0739	648.31	.1046	716.42	.1192	3.865±0.055
	16×16	**371.06**	.0938	277.01	.0723	611.18	.1023	**704.96**	.1163	3.911±0.038
	32×32	378.04	**.0890**	**272.09**	**.0671**	**601.83**	**.0957**	711.90	**.1098**	3.914±0.020

Grid denotes the pooled output resolution; 16×16 is the reference setting. Bold indicates the best prediction metric within each backbone–split pair, ranked before rounding; leading zeros are omitted from LPIPS. Correction latency excludes backbone inference and is measured on the first 512 Standard clips, batch size 1, on the same RTX 3090 as the earlier efficiency study. Values are means ± sample standard deviations of three complete timing runs.

Table 6: Per-clip MSE on the ten laboratory video clips.

Clip	Source	δ-Adapter	+ Regional	STORC
01	989.27	989.27	279.49	238.34
02	793.78	793.78	247.95	191.67
03	783.80	783.80	252.16	202.77
04	863.98	863.98	292.79	246.18
05	873.30	873.06	303.04	262.94
06	625.05	624.91	240.50	198.01
07	906.50	906.28	280.80	247.91
08	584.63	584.42	218.98	193.89
09	794.83	794.25	250.16	208.94
10	546.89	546.53	194.36	144.71
Mean	776.20	776.03	256.02	213.54

A.8 Laboratory Video Transfer

This pilot evaluation uses ten laboratory video clips and is separate from the BridgeData test splits. Each 30 FPS clip is sampled at source-frame indices $0, 5, \ldots, 55$, producing twelve frames at 6 FPS, and resized from 960×720 to 160×120. The first two frames are inputs and the remaining ten are targets. All methods share the exact cached raw predictions of the frozen BridgeData-trained TAU. STORC and the + Regional variant use feedback at $p = 2, 4, 6, 8$.

The output-only δ-Adapter is initialized to the identity and processed in clip order with batch sizes $4, 4, 2$. It retains state across batches, performs one Adam step per reveal with learning rate 10^{-4} and gradient clipping at 0.01, and uses an additive correction bound of 0.1. A full-history update after each batch affects only later batches. MSE is computed on floating-point predictions before image export, summed over spatial and RGB dimensions and averaged over the ten target frames.

A.9 Complete Prediction Results

Tables 7–11 report all six metrics for the 45 backbone–split combinations under $2 \rightarrow 10$ forecasting. Each table covers one backbone and the nine test splits, using the same split abbreviations as Table 1. MSE, MAE, LPIPS, and FVD are lower-is-better; SSIM and PSNR are higher-is-better. Best and second-best values are determined before rounding among Source, the five adaptation baselines, and STORC. Reforecast is listed separately as an additional higher-compute reference and is excluded from these rankings. Leading zeros are omitted from LPIPS values, as in Table 1. The overall average MSE reduction assigns equal weight to the 45 setting-wise relative reductions rather than pooling raw errors across datasets.

Table 7: Complete prediction results with PredRNNv2 under $2 \rightarrow 10$ forecasting. Bold and underlined values indicate the best and second-best results among the seven main methods, excluding the additional Reforecast reference.

Metric	Split	Source	TAFAS	PETSA	ST-TTC	COSA	δ-Adapter	STORC	Reforecast
MSE ↓	B-Std.	519.6	519.1	526.5	520.0	520.0	514.7	**375.3**	318.4
	B-Task	488.2	484.7	481.0	485.2	486.0	476.1	**271.1**	222.3
	B-Scene	1294.8	1265.8	1132.8	1288.1	1274.5	1221.1	**623.5**	513.0
	B-Both	1203.8	1132.6	1065.5	1196.0	1166.9	1098.0	**722.2**	568.0
	R-Std.	54.1	54.1	54.4	54.1	54.1	54.1	**40.0**	36.7
	R-Dyn.	136.6	135.5	135.0	136.5	136.7	135.9	**76.5**	74.7
	R-Scene	98.5	98.4	98.0	98.3	98.5	97.9	**75.3**	70.4
	R-Joint	324.0	323.1	323.8	323.6	323.5	320.1	**194.5**	189.1
	BAIR	284.9	284.8	285.9	284.8	284.9	284.9	**202.4**	183.4
MAE ↓	B-Std.	2610.5	2614.2	2683.5	2605.1	2615.7	2617.0	**1980.2**	1831.5
	B-Task	2340.3	2365.2	2388.0	2331.3	2348.8	2341.7	**1646.0**	1508.1
	B-Scene	4699.0	4596.8	4220.8	4659.5	4668.9	4513.4	**2942.5**	2514.3
	B-Both	4624.9	4500.6	4142.7	4578.5	4594.1	4357.5	**3171.5**	2700.8
	R-Std.	364.4	362.4	369.3	361.1	364.2	363.3	**275.2**	268.4
	R-Dyn.	711.6	704.1	691.3	707.0	711.9	719.2	**466.2**	459.9
	R-Scene	589.6	583.4	583.4	581.6	588.1	582.5	**451.7**	450.9
	R-Joint	1295.5	1302.6	1280.8	1291.9	1304.8	1309.6	**887.0**	878.8
	BAIR	1066.4	1066.9	1068.1	1059.9	1066.5	1064.7	**765.3**	735.0
SSIM ↑	B-Std.	0.8235	0.8235	0.8217	0.8236	0.8234	0.8241	**0.8555**	0.8635
	B-Task	0.8558	0.8558	0.8549	0.8566	0.8562	0.8561	**0.8868**	0.8951
	B-Scene	0.7551	0.7576	0.7725	0.7556	0.7572	0.7624	**0.8221**	0.8332
	B-Both	0.7395	0.7473	0.7620	0.7404	0.7435	0.7525	**0.8016**	0.8149
	R-Std.	0.8850	0.8851	0.8848	0.8852	0.8850	0.8849	**0.9150**	0.9196
	R-Dyn.	0.7407	0.7441	0.7509	0.7410	0.7406	0.7409	**0.8249**	0.8279
	R-Scene	0.7937	0.7940	0.7951	0.7942	0.7937	0.7957	**0.8370**	0.8389
	R-Joint	0.5593	0.5600	0.5651	0.5598	0.5590	0.5591	**0.6796**	0.6821
	BAIR	0.6821	0.6823	0.6806	0.6825	0.6821	0.6819	**0.7661**	0.7720
PSNR ↑	B-Std.	22.47	22.48	22.37	22.47	22.46	22.50	**24.09**	24.73
	B-Task	22.29	22.32	22.32	22.34	22.32	22.39	**24.64**	25.54
	B-Scene	17.59	17.68	18.10	17.61	17.66	17.86	**20.63**	21.64
	B-Both	18.01	18.30	18.68	18.05	18.16	18.47	**20.37**	21.44
	R-Std.	25.76	25.77	25.71	25.78	25.76	25.76	**27.55**	28.39
	R-Dyn.	21.18	21.26	21.41	21.19	21.17	21.19	**24.09**	24.24
	R-Scene	21.94	21.96	21.98	21.96	21.95	21.99	**23.21**	23.43
	R-Joint	16.59	16.60	16.65	16.60	16.59	16.63	**19.06**	19.12
	BAIR	16.91	16.91	16.90	16.92	16.91	16.91	**18.87**	19.12
LPIPS ↓	B-Std.	.1376	.1373	.1388	.1374	.1376	.1361	**.0919**	.1024
	B-Task	.1079	.1084	.1088	.1079	.1079	.1076	**.0696**	.0802
	B-Scene	.1737	.1718	.1510	.1734	.1722	.1666	**.1033**	.1154
	B-Both	.1984	.1909	.1730	.1979	.1946	.1854	**.1215**	.1341
	R-Std.	.0561	.0561	.0562	.0561	.0561	.0562	**.0364**	.0351
	R-Dyn.	.1090	.1083	.1034	.1090	.1091	.1083	**.0635**	.0647
	R-Scene	.1104	.1103	.1084	.1104	.1105	.1086	**.0694**	.0845
	R-Joint	.1996	.1992	.1922	.1996	.1998	.2000	**.1198**	.1369
	BAIR	.2288	.2287	.2283	.2286	.2288	.2283	**.1155**	.1542
FVD ↓	B-Std.	401.4	409.0	413.8	403.3	403.3	409.9	**120.2**	178.1
	B-Task	571.9	571.9	569.6	572.7	571.3	573.4	**232.8**	282.7
	B-Scene	968.7	947.7	880.2	968.4	976.6	961.3	**529.7**	378.0
	B-Both	746.5	717.6	698.3	749.7	780.0	731.6	**364.1**	346.8
	R-Std.	464.4	465.6	466.1	464.3	464.9	465.5	**218.4**	179.0
	R-Dyn.	582.1	579.4	553.3	581.7	579.5	582.2	**170.2**	140.0
	R-Scene	868.9	874.2	862.7	868.7	865.5	878.5	**246.7**	417.7
	R-Joint	1396.3	1379.3	1391.9	1395.3	1388.1	1388.9	**462.1**	594.9
	BAIR	1331.1	1334.6	1318.2	1330.8	1330.4	1333.5	**493.8**	699.0

Table 8: Complete prediction results with TAU under $2 \rightarrow 10$ forecasting. Bold and underlined values indicate the best and second-best results among the seven main methods, excluding the additional Reforecast reference.

Metric	Split	Source	TAFAS	PETSA	ST-TTC	COSA	δ-Adapter	STORC	Reforecast
MSE ↓	B-Std.	479.1	475.4	484.3	479.6	479.4	468.4	**356.3**	280.0
	B-Task	618.0	587.6	523.9	611.3	611.1	576.1	**290.5**	218.8
	B-Scene	1325.2	1313.4	1181.5	1320.4	1286.9	1229.6	**627.0**	531.8
	B-Both	1256.7	1196.7	1076.2	1240.3	1183.1	1121.0	**720.2**	570.5
	R-Std.	58.3	57.7	57.7	58.0	57.6	58.1	**41.9**	39.2
	R-Dyn.	139.8	139.1	139.4	139.3	139.0	138.8	**77.5**	77.9
	R-Scene	101.8	100.9	100.5	101.4	100.8	100.5	**77.1**	76.2
	R-Joint	319.8	317.9	318.9	319.6	317.5	316.6	**195.3**	196.9
	BAIR	287.1	286.9	289.0	286.9	287.0	286.9	**203.3**	185.0
MAE ↓	B-Std.	2455.4	2453.9	2545.3	2444.4	2459.1	2440.3	**1917.1**	1713.1
	B-Task	2611.8	2605.3	2538.1	2594.2	2628.4	2560.7	**1713.9**	1461.1
	B-Scene	4783.3	4610.1	4290.6	4704.6	4583.5	4602.4	**2912.3**	2624.3
	B-Both	4738.3	4516.0	4131.4	4609.0	4522.0	4380.7	**3131.2**	2802.8
	R-Std.	417.4	404.2	405.6	411.9	405.4	413.0	**303.4**	301.8
	R-Dyn.	755.1	744.6	726.2	748.4	747.0	751.5	**488.2**	507.6
	R-Scene	630.0	611.5	606.3	621.9	615.3	615.3	**479.9**	527.5
	R-Joint	1278.0	1286.3	1276.6	1274.9	1287.9	1301.7	**889.9**	907.7
	BAIR	1085.7	1080.7	1085.6	1075.9	1083.7	1083.5	**785.0**	779.2
SSIM ↑	B-Std.	0.8307	0.8311	0.8280	0.8308	0.8306	0.8320	**0.8575**	0.8699
	B-Task	0.8499	0.8513	0.8514	0.8509	0.8503	0.8512	**0.8840**	0.8977
	B-Scene	0.7644	0.7651	0.7728	0.7650	0.7655	0.7696	**0.8247**	0.8370
	B-Both	0.7532	0.7578	0.7695	0.7544	0.7578	0.7616	**0.8054**	0.8186
	R-Std.	0.8776	0.8778	0.8773	0.8777	0.8778	0.8774	**0.9102**	0.9150
	R-Dyn.	0.7402	0.7407	0.7454	0.7408	0.7406	0.7409	**0.8243**	0.8304
	R-Scene	0.7883	0.7891	0.7904	0.7889	0.7887	0.7920	**0.8333**	0.8336
	R-Joint	0.5624	0.5620	0.5646	0.5630	0.5620	0.5589	**0.6771**	0.6800
	BAIR	0.6769	0.6771	0.6754	0.6774	0.6769	0.6773	**0.7628**	0.7675
PSNR ↑	B-Std.	22.88	22.90	22.74	22.88	22.87	22.95	**24.30**	25.13
	B-Task	21.44	21.65	21.90	21.51	21.49	21.67	**24.31**	25.51
	B-Scene	17.43	17.52	17.94	17.46	17.57	17.75	**20.57**	21.40
	B-Both	17.88	18.17	18.63	17.96	18.14	18.38	**20.38**	21.35
	R-Std.	25.05	25.14	25.10	25.08	25.12	25.08	**26.98**	27.46
	R-Dyn.	20.98	21.06	21.17	21.02	21.04	21.04	**23.95**	23.80
	R-Scene	21.73	21.81	21.82	21.76	21.80	21.82	**23.06**	22.96
	R-Joint	16.66	16.67	16.69	16.66	16.67	16.68	**19.05**	18.98
	BAIR	16.85	16.86	16.82	16.86	16.86	16.87	**18.82**	18.99
LPIPS ↓	B-Std.	.1412	.1397	.1427	.1412	.1413	.1375	**.0925**	.0993
	B-Task	.1161	.1161	.1151	.1161	.1161	.1137	**.0720**	.0799
	B-Scene	.1655	.1655	.1548	.1653	.1645	.1594	**.0992**	.1106
	B-Both	.1834	.1793	.1642	.1830	.1800	.1759	**.1163**	.1271
	R-Std.	.0594	.0591	.0593	.0592	.0593	.0592	**.0382**	.0361
	R-Dyn.	.1033	.1031	.1017	.1032	.1031	.1018	**.0613**	.0579
	R-Scene	.1093	.1095	.1093	.1092	.1096	.1063	**.0686**	.0765
	R-Joint	.1860	.1867	.1857	.1859	.1865	.1875	**.1162**	.1172
	BAIR	.2334	.2333	.2334	.2333	.2334	.2331	**.1177**	.1601
FVD ↓	B-Std.	364.4	370.6	365.5	361.8	364.5	372.2	**111.9**	186.3
	B-Task	545.7	543.3	554.3	545.7	542.9	547.6	**233.7**	290.5
	B-Scene	727.4	723.0	649.6	728.2	726.8	719.0	**399.0**	385.7
	B-Both	659.4	672.6	643.0	655.0	684.3	670.7	**300.7**	385.5
	R-Std.	518.4	506.2	511.3	516.1	513.2	513.4	**229.5**	271.1
	R-Dyn.	597.6	575.7	585.0	594.7	578.1	597.4	**190.7**	167.4
	R-Scene	948.7	920.2	930.2	943.5	920.6	948.8	**252.9**	407.0
	R-Joint	1527.6	1498.0	1532.7	1525.5	1505.0	1538.1	**445.4**	518.1
	BAIR	1409.7	1420.6	1416.3	1412.1	1409.0	1395.7	**482.1**	800.2

Table 9: Complete prediction results with SimVPv2 under $2 \rightarrow 10$ forecasting. Bold and underlined values indicate the best and second-best results among the seven main methods, excluding the additional Reforecast reference.

Metric	Split	Source	TAFAS	PETSA	ST-TTC	COSA	δ-Adapter	STORC	Reforecast
MSE ↓	B-Std.	533.6	532.3	546.1	534.0	534.2	528.7	**382.3**	329.7
	B-Task	493.3	485.9	489.2	487.6	489.3	481.3	**269.0**	230.8
	B-Scene	1166.8	1160.4	1105.6	1166.5	1163.6	1122.5	**600.9**	516.2
	B-Both	1069.9	1057.8	1041.3	1064.9	1060.7	1016.6	**688.2**	570.1
	R-Std.	58.1	57.9	58.5	58.0	58.1	58.0	**42.1**	39.8
	R-Dyn.	141.4	140.6	140.9	140.7	140.9	140.2	**78.5**	80.8
	R-Scene	102.8	102.0	101.5	102.2	102.1	101.4	**77.6**	78.4
	R-Joint	322.3	321.8	323.1	322.0	321.8	319.6	**196.6**	198.9
	BAIR	287.5	287.0	290.5	287.2	287.3	287.3	**203.8**	186.9
MAE ↓	B-Std.	2775.8	2745.5	2862.0	2759.5	2780.3	2731.4	**2054.2**	2030.6
	B-Task	2510.1	2492.9	2536.3	2493.9	2501.7	2474.2	**1695.6**	1621.0
	B-Scene	4222.6	4201.8	4114.6	4203.8	4221.8	4152.4	**2784.7**	2653.1
	B-Both	4255.8	4186.8	4079.1	4189.9	4232.7	4092.9	**3006.5**	2911.7
	R-Std.	407.3	401.8	411.4	404.4	406.9	403.2	**302.4**	307.2
	R-Dyn.	767.1	754.0	738.1	759.3	758.4	757.7	**496.9**	537.6
	R-Scene	640.9	620.2	613.0	630.0	624.7	624.7	**486.5**	549.0
	R-Joint	1288.4	1292.5	1281.2	1283.9	1292.5	1316.1	**897.7**	928.6
	BAIR	1098.6	1081.4	1090.3	1081.9	1088.0	1093.3	**789.7**	792.8
SSIM ↑	B-Std.	0.8191	0.8202	0.8164	0.8194	0.8190	0.8220	**0.8525**	0.8593
	B-Task	0.8530	0.8540	0.8524	0.8541	0.8538	0.8544	**0.8852**	0.8925
	B-Scene	0.7760	0.7762	0.7780	0.7762	0.7759	0.7787	**0.8281**	0.8351
	B-Both	0.7638	0.7649	0.7680	0.7646	0.7644	0.7690	**0.8093**	0.8153
	R-Std.	0.8757	0.8759	0.8752	0.8758	0.8757	0.8760	**0.9092**	0.9137
	R-Dyn.	0.7366	0.7369	0.7386	0.7371	0.7367	0.7386	**0.8227**	0.8260
	R-Scene	0.7850	0.7859	0.7891	0.7857	0.7856	0.7894	**0.8319**	0.8308
	R-Joint	0.5603	0.5601	0.5609	0.5609	0.5601	0.5564	**0.6754**	0.6766
	BAIR	0.6769	0.6775	0.6744	0.6775	0.6770	0.6770	**0.7622**	0.7662
PSNR ↑	B-Std.	22.22	22.26	22.08	22.23	22.21	22.31	**23.94**	24.36
	B-Task	22.10	22.19	22.15	22.17	22.15	22.22	**24.62**	25.23
	B-Scene	17.90	17.94	18.19	17.91	17.92	18.10	**20.76**	21.43
	B-Both	18.48	18.59	18.73	18.52	18.54	18.77	**20.62**	21.22
	R-Std.	25.06	25.09	25.01	25.08	25.06	25.09	**26.95**	27.40
	R-Dyn.	20.91	21.00	21.06	20.96	20.98	21.00	**23.88**	23.54
	R-Scene	21.66	21.73	21.77	21.70	21.72	21.76	**23.02**	22.81
	R-Joint	16.63	16.64	16.64	16.64	16.63	16.64	**19.02**	18.90
	BAIR	16.83	16.85	16.79	16.84	16.84	16.85	**18.80**	18.93
LPIPS ↓	B-Std.	.1392	.1385	.1407	.1392	.1393	.1381	**.0918**	.0989
	B-Task	.1100	.1100	.1100	.1100	.1100	.1084	**.0699**	.0780
	B-Scene	.1549	.1545	.1503	.1547	.1545	.1518	**.0969**	.1087
	B-Both	.1734	.1724	.1660	.1730	.1726	.1688	**.1124**	.1257
	R-Std.	.0608	.0608	.0611	.0607	.0608	.0604	**.0387**	.0367
	R-Dyn.	.1014	.1015	.1054	.1013	.1010	.1002	**.0611**	.0589
	R-Scene	.1076	.1079	.1050	.1075	.1076	.1049	**.0685**	.0769
	R-Joint	.1797	.1800	.1840	.1796	.1795	.1825	**.1156**	.1152
	BAIR	.2311	.2307	.2270	.2311	.2309	.2300	**.1176**	.1589
FVD ↓	B-Std.	406.3	409.3	401.7	407.0	405.5	402.7	**108.2**	177.5
	B-Task	590.2	586.5	586.9	590.5	586.7	593.8	**217.3**	283.0
	B-Scene	739.7	728.8	684.7	735.8	731.6	725.7	**372.9**	406.2
	B-Both	677.3	655.2	653.4	670.0	665.2	670.0	**264.2**	389.6
	R-Std.	524.7	514.0	520.6	522.1	521.8	521.7	**228.4**	261.7
	R-Dyn.	599.0	569.1	569.7	595.8	574.5	599.4	**200.4**	167.1
	R-Scene	986.9	958.2	945.9	981.3	953.7	990.0	**259.1**	391.2
	R-Joint	1585.4	1573.7	1562.9	1583.4	1570.8	1581.5	**460.8**	488.1
	BAIR	1418.1	1430.0	1408.3	1419.7	1422.2	1419.0	**506.7**	815.6

Table 10: Complete prediction results with PredFormer under $2 \rightarrow 10$ forecasting. Bold and underlined values indicate the best and second-best results among the seven main methods, excluding the additional Reforecast reference.

Metric	Split	Source	TAFAS	PETSA	ST-TTC	COSA	δ-Adapter	STORC	Reforecast
MSE ↓	B-Std.	525.1	521.2	529.9	525.7	525.5	517.1	**376.2**	320.3
	B-Task	571.9	563.3	522.8	565.2	567.2	550.8	**293.2**	247.0
	B-Scene	1359.9	1302.4	1204.1	1356.1	1338.7	1260.5	**635.3**	548.2
	B-Both	1270.9	1201.0	1089.2	1261.3	1184.3	1139.1	**739.0**	597.3
	R-Std.	58.7	58.7	59.1	58.7	58.7	58.6	**42.1**	39.1
	R-Dyn.	143.2	142.9	143.5	142.7	143.1	141.5	**78.5**	77.5
	R-Scene	108.2	107.9	107.5	107.9	108.1	105.9	**78.1**	75.3
	R-Joint	340.9	337.4	336.3	340.6	334.9	329.9	**198.6**	194.9
	BAIR	293.8	293.8	293.0	293.7	293.9	293.6	**206.5**	193.5
MAE ↓	B-Std.	2741.8	2738.8	2804.0	2739.3	2746.8	2730.4	**2100.9**	1984.3
	B-Task	2705.9	2711.0	2672.2	2692.2	2712.5	2691.1	**1856.0**	1746.1
	B-Scene	4601.1	4538.2	4313.9	4575.1	4638.7	4494.5	**2929.8**	2597.6
	B-Both	4637.6	4518.0	4233.2	4585.1	4523.2	4385.6	**3205.3**	2839.8
	R-Std.	397.6	398.1	407.9	396.3	398.1	399.5	**301.1**	292.7
	R-Dyn.	772.2	769.4	768.2	766.0	770.5	765.1	**500.5**	489.1
	R-Scene	643.6	641.4	643.3	637.4	641.9	639.6	**486.1**	483.8
	R-Joint	1271.0	1286.0	1288.7	1269.2	1303.1	1309.2	**883.9**	864.5
	BAIR	1116.5	1118.8	1113.7	1111.4	1117.6	1117.2	**805.1**	800.1
SSIM ↑	B-Std.	0.8000	0.8011	0.7979	0.8002	0.7999	0.8026	**0.8422**	0.8439
	B-Task	0.8272	0.8278	0.8292	0.8282	0.8276	0.8290	**0.8720**	0.8734
	B-Scene	0.7392	0.7401	0.7486	0.7402	0.7394	0.7444	**0.8114**	0.8164
	B-Both	0.7263	0.7327	0.7415	0.7278	0.7342	0.7373	**0.7907**	0.7954
	R-Std.	0.8728	0.8728	0.8719	0.8731	0.8728	0.8724	**0.9082**	0.9118
	R-Dyn.	0.7221	0.7221	0.7231	0.7238	0.7219	0.7252	**0.8165**	0.8184
	R-Scene	0.7745	0.7745	0.7735	0.7757	0.7745	0.7788	**0.8278**	0.8258
	R-Joint	0.5569	0.5563	0.5531	0.5578	0.5541	0.5518	**0.6735**	0.6776
	BAIR	0.6585	0.6586	0.6576	0.6605	0.6585	0.6578	**0.7541**	0.7498
PSNR ↑	B-Std.	22.19	22.22	22.11	22.19	22.19	22.25	**23.85**	24.28
	B-Task	21.58	21.64	21.81	21.64	21.62	21.70	**24.17**	24.78
	B-Scene	17.36	17.53	17.87	17.37	17.40	17.64	**20.49**	21.27
	B-Both	17.74	18.08	18.49	17.78	18.06	18.22	**20.19**	21.05
	R-Std.	25.11	25.11	25.02	25.12	25.11	25.11	**26.98**	27.52
	R-Dyn.	20.82	20.84	20.84	20.85	20.83	20.90	**23.82**	23.82
	R-Scene	21.47	21.49	21.48	21.49	21.48	21.57	**22.98**	23.07
	R-Joint	16.49	16.51	16.50	16.49	16.49	16.55	**19.00**	19.06
	BAIR	16.76	16.76	16.77	16.77	16.76	16.76	**18.74**	18.80
LPIPS ↓	B-Std.	.1365	.1346	.1373	.1364	.1365	.1328	**.0923**	.0959
	B-Task	.1082	.1078	.1057	.1081	.1082	.1063	**.0703**	.0721
	B-Scene	.1627	.1597	.1478	.1624	.1622	.1576	**.1008**	.1000
	B-Both	.1830	.1727	.1615	.1826	.1757	.1734	**.1169**	.1203
	R-Std.	.0568	.0567	.0569	.0567	.0568	.0565	**.0374**	.0340
	R-Dyn.	.0987	.0986	.0977	.0986	.0986	.0979	**.0604**	.0587
	R-Scene	.1026	.1022	.1004	.1024	.1023	.1002	**.0673**	.0756
	R-Joint	.1759	.1758	.1785	.1758	.1772	.1787	**.1153**	.1182
	BAIR	.2311	.2310	.2303	.2309	.2310	.2311	**.1165**	.1543
FVD ↓	B-Std.	360.3	363.2	365.4	358.7	360.0	365.0	**112.9**	133.3
	B-Task	568.4	564.4	583.1	568.8	567.1	573.9	**242.4**	236.9
	B-Scene	663.1	658.7	636.1	663.9	659.9	649.4	**358.6**	313.1
	B-Both	604.5	617.8	605.4	604.2	638.4	613.8	**286.9**	275.6
	R-Std.	507.3	505.7	508.4	505.9	506.0	509.3	**232.1**	248.3
	R-Dyn.	649.7	650.0	632.7	645.4	647.6	650.8	**196.2**	164.4
	R-Scene	987.1	970.4	944.9	977.8	969.4	991.2	**277.4**	411.6
	R-Joint	1551.4	1514.6	1521.2	1546.8	1505.3	1540.7	**462.8**	514.2
	BAIR	1456.5	1459.7	1451.9	1447.2	1456.8	1445.1	**486.8**	748.3

Table 11: Complete prediction results with PFGNet under $2 \rightarrow 10$ forecasting. Bold and underlined values indicate the best and second-best results among the seven main methods, excluding the additional Reforecast reference.

Metric	Split	Source	TAFAS	PETSA	ST-TTC	COSA	δ-Adapter	STORC	Reforecast
MSE ↓	B-Std.	509.4	507.4	516.4	509.9	509.9	502.3	**371.1**	307.3
	B-Task	516.0	508.1	495.4	517.2	517.2	497.5	**277.0**	223.1
	B-Scene	1245.0	1227.3	1159.6	1240.2	1233.0	1173.1	**611.2**	514.1
	B-Both	1152.4	1121.2	1066.9	1143.6	1139.4	1072.4	**705.0**	576.1
	R-Std.	56.5	56.4	56.7	56.4	56.5	56.3	**41.2**	38.1
	R-Dyn.	147.3	143.2	142.3	145.6	142.7	144.8	**78.9**	83.1
	R-Scene	105.1	104.2	104.1	104.6	104.2	103.7	**77.3**	75.4
	R-Joint	321.6	319.2	318.8	320.9	316.2	318.9	**195.2**	192.1
	BAIR	287.2	287.0	288.5	287.0	287.1	287.0	**203.4**	184.3
MAE ↓	B-Std.	2565.8	2562.5	2657.5	2548.4	2567.9	2542.2	**1974.6**	1806.5
	B-Task	2394.2	2392.5	2467.4	2393.0	2404.7	2369.0	**1672.6**	1484.1
	B-Scene	4537.2	4457.1	4219.6	4474.5	4420.7	4399.3	**2858.9**	2529.7
	B-Both	4447.3	4251.4	4012.9	4330.3	4360.6	4223.7	**3046.7**	2757.4
	R-Std.	395.6	392.3	400.3	393.1	395.0	387.9	**292.4**	280.0
	R-Dyn.	846.6	797.5	754.1	831.3	797.4	813.2	**518.7**	582.0
	R-Scene	671.3	659.4	656.4	664.9	660.3	656.9	**486.3**	516.0
	R-Joint	1308.7	1303.1	1296.5	1302.9	1295.4	1320.7	**900.2**	922.8
	BAIR	1072.7	1073.5	1077.8	1066.8	1072.5	1072.9	**775.5**	743.6
SSIM ↑	B-Std.	0.8220	0.8227	0.8202	0.8227	0.8223	0.8246	**0.8538**	0.8629
	B-Task	0.8527	0.8536	0.8515	0.8534	0.8532	0.8540	**0.8848**	0.8943
	B-Scene	0.7645	0.7657	0.7721	0.7656	0.7658	0.7701	**0.8247**	0.8353
	B-Both	0.7587	0.7630	0.7696	0.7602	0.7603	0.7652	**0.8071**	0.8169
	R-Std.	0.8773	0.8776	0.8778	0.8778	0.8777	0.8784	**0.9108**	0.9160
	R-Dyn.	0.7384	0.7399	0.7430	0.7397	0.7400	0.7426	**0.8247**	0.8272
	R-Scene	0.7856	0.7859	0.7889	0.7865	0.7864	0.7891	**0.8332**	0.8339
	R-Joint	0.5644	0.5644	0.5639	0.5653	0.5649	0.5625	**0.6784**	0.6803
	BAIR	0.6779	0.6779	0.6768	0.6794	0.6789	0.6778	**0.7633**	0.7706
PSNR ↑	B-Std.	22.52	22.55	22.40	22.53	22.52	22.60	**24.09**	24.72
	B-Task	22.15	22.23	22.19	22.17	22.16	22.27	**24.56**	25.47
	B-Scene	17.63	17.70	18.04	17.65	17.68	17.91	**20.66**	21.48
	B-Both	18.24	18.44	18.71	18.29	18.29	18.60	**20.52**	21.30
	R-Std.	25.26	25.28	25.21	25.28	25.26	25.30	**27.11**	27.78
	R-Dyn.	20.44	20.75	20.96	20.53	20.75	20.64	**23.72**	23.10
	R-Scene	21.51	21.58	21.59	21.54	21.57	21.61	**23.03**	23.00
	R-Joint	16.62	16.65	16.65	16.63	16.68	16.64	**19.02**	18.98
	BAIR	16.87	16.87	16.84	16.87	16.87	16.87	**18.82**	19.05
LPIPS ↓	B-Std.	.1442	.1430	.1465	.1444	.1443	.1424	**.0938**	.1053
	B-Task	.1154	.1153	.1183	.1156	.1156	.1135	**.0723**	.0830
	B-Scene	.1741	.1734	.1567	.1740	.1738	.1673	**.1023**	.1165
	B-Both	.1854	.1816	.1717	.1853	.1850	.1782	**.1163**	.1327
	R-Std.	.0643	.0640	.0643	.0641	.0642	.0635	**.0400**	.0396
	R-Dyn.	.1105	.1103	.1099	.1103	.1102	.1090	**.0634**	.0676
	R-Scene	.1206	.1207	.1197	.1204	.1210	.1179	**.0726**	.0903
	R-Joint	.1932	.1926	.1950	.1929	.1932	.1931	**.1191**	.1318
	BAIR	.2326	.2324	.2318	.2322	.2322	.2330	**.1179**	.1609
FVD ↓	B-Std.	393.6	396.9	395.7	390.3	392.2	396.7	**107.3**	204.8
	B-Task	567.4	567.4	580.0	566.6	565.9	574.2	**222.9**	305.4
	B-Scene	735.6	737.8	633.1	735.1	728.6	711.6	**368.1**	395.3
	B-Both	622.0	621.1	611.5	620.2	623.5	623.5	**269.1**	378.1
	R-Std.	498.0	500.9	501.1	498.1	498.6	501.6	**221.0**	264.7
	R-Dyn.	565.5	554.0	568.8	563.0	557.8	567.1	**178.4**	195.6
	R-Scene	945.1	913.6	931.5	946.2	927.1	944.0	**249.3**	466.3
	R-Joint	1525.0	1480.7	1509.7	1522.8	1509.6	1530.2	**429.2**	548.5
	BAIR	1360.1	1357.2	1349.0	1363.1	1352.3	1365.0	**496.7**	734.9