# LIFT-JEPA: TASK-AWARE JOINT-EMBEDDING SUPERVISION VIA TARGET INTERVENTIONS FOR SPATIOTEMPORAL FORECASTING

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

## ABSTRACT

In spatiotemporal forecasting where task-critical phenomena are sparse, prediction losses applied in observation space allow pixel-wise errors in task-critical regions to be directly reweighted. By contrast, joint-embedding predictive learning applies the loss in representation space; because latent units lack an explicit correspondence to task semantics expressed in observation space, observation-space task weights cannot be directly applied to the representation-matching loss. To this end, we introduce LIFT-JEPA (**L**atent **I**mportance **F**rom **T**arget Interventions for Joint-Embedding Prediction). Given a future observation, LIFT-JEPA constructs an intervened counterpart by removing task-critical structures and encodes both using the same target encoder. The resulting representation differences are used to estimate the task relevance of individual representation units and reweight their matching losses, thereby translating task priorities specified in observation space into representation-space supervision. We evaluate LIFT-JEPA across precipitation nowcasting and urban traffic flow prediction using three forecasting backbones with distinct inductive biases. Compared with Vanilla JEPA and alternative task-weighting strategies, LIFT-JEPA consistently improves task-relevant forecasting performance across both domains, from intensity-defined targets in radar nowcasting to temporal-drift targets in traffic flow prediction. These results demonstrate that LIFT-JEPA can translate task semantics ranging from spatial intensity structures to temporal drift into effective representation-space supervision.

## 1 INTRODUCTION

Spatiotemporal forecasting infers future states from historical observations and supports applications in weather, traffic, autonomous driving, and remote sensing. In many tasks, task-critical phenomena occupy only a small fraction of the observation field, allowing prevalent background regions to dominate the aggregate gradient under uniformly averaged pixelwise losses. Radar echo extrapolation exemplifies this imbalance: intense echoes relevant to heavy-rainfall warnings are sparse among prevalent no-echo and weak-echo regions, often resulting in intensity underestimation and missed detection. A similar issue arises in traffic-flow forecasting, where large temporal changes are spatially localized. Both settings call for task-aware supervision compatible with latent representation learning.

Spatiotemporal forecasting, including radar nowcasting and traffic flow forecasting, has advanced through recurrent, physics-guided, feedforward convolutional, attention-based, and frequency-domain architectures (Shi et al., 2015; Wang et al., 2017; Guen & Thome, 2020; Gao et al., 2022a; Tan et al., 2023a; Gao et al., 2022b; Lin et al., 2025). These approaches are typically trained by minimizing pixelwise losses on decoded predictions in observation space, where each local loss term corresponds to a particular pixel or grid cell in the target. This direct correspondence allows task importance to be expressed through loss weighting. In radar nowcasting, for example, B-MSE, B-MAE, and hybrid weighted losses assign larger weights to pixels with higher ground-truth intensities (Shi et al., 2017; Cao et al., 2022). However, when the training loss is computed in representation space rather than observation space, task importance defined over pixels or grid cells does not directly specify how individual representation elements should be weighted.



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.

Joint-embedding predictive learning predicts target representations from context representations and has been applied to masked or future representation learning in images, videos, and sequential data. In sequential tasks, time naturally separates historical context from future targets (Assran et al., 2023; Bardes et al., 2024; Schwarzer et al., 2020; Assran et al., 2025). Standard JEPA objectives typically aggregate matching errors uniformly across representation units. Existing extensions shift the learning focus using model-internal features, predefined geometry, or latent-variable masks (Mo & Yun, 2024; He et al., 2025; Mur-Labadia et al., 2026; Nam et al., 2026), whereas perturbation responses are primarily used for post-hoc interpretation (Fong & Vedaldi, 2017; Petsiuk et al., 2018). This leaves unresolved how forecasting events defined in observation space can determine matching-loss weights for spatial–channel representation units during training. Directly pooling observation masks provides only coarse geometric alignment and cannot capture how receptive-field expansion and channel mixing redistribute task information across latent units. An encoder-conditioned mapping from observation-space task semantics to differentiated representation supervision is therefore needed.

To this end, we introduce **LIFT-JEPA**. Although the semantics of individual latent units may be opaque, their responses to a task-specific intervention can be measured. Given a factual future, LIFT-JEPA removes designated task-critical structures in observation space and passes the factual and intervened futures through the same exponential-moving-average target encoder. Their representation difference estimates task sensitivity and is normalized into sample- and time-adaptive spatial–channel importance weights. LIFT-JEPA retains the standard full-field representation-matching term and uses these weights to construct an additional task-focused term between the predicted representation and the factual target representation. The intervened future only locates units that deserve greater emphasis; it is not used as the matching target. This training-only branch incurs no inference-time overhead.

We evaluate LIFT-JEPA on two regional radar datasets and TaxiBJ using PhyDNet, SimVP, and TAU, covering physics-guided recurrent, feedforward convolutional, and attention-based designs. Across all six combinations of radar datasets and backbones, LIFT-JEPA improves CSI-M and CSI at the intervention threshold over standard JEPA. Controlled comparisons indicate that the gains depend on the semantics of the intervention and the placement of the resulting weights, rather than increased loss scale or direct mask pooling. Analyses across reflectivity thresholds further characterize the associated trade-off between detection and false alarms. On TaxiBJ, redefining the intervention to target locations with large future changes reduces MSE within those regions, indicating that the mechanism extends beyond radar intensity. Together, these results support the use of target encoder responses to translate task definitions in observation space into loss weights in representation space.

Our main contributions are summarized as follows:

- We identify a task-supervision transfer gap in joint-embedding forecasting: observation-space task preferences do not directly map to matching-loss weights for spatial–channel representation units.
- We propose LIFT-JEPA, which converts target-encoder response differences between factual and intervened futures into sample-adaptive spatial–channel weights that strengthen matching to the factual target representation.
- We validate LIFT-JEPA on two radar datasets and TaxiBJ with three diverse backbones, showing gains in the targeted regions and isolating the effects of intervention semantics and weight placement through controlled comparisons.

## 2 Related Work

### 2.1 Spatiotemporal Forecasting and Observation-Space Loss Weighting

Spatiotemporal forecasting predicts future spatial fields from historical observations in domains such as radar nowcasting and urban traffic flow forecasting. Existing deterministic backbones model spatiotemporal dynamics using different inductive biases. ConvLSTM (Shi et al., 2015) and PredRNN (Wang et al., 2017) employ recurrent state updates, while PhyDNet (Guen & Thome, 2020) uses parallel PhyCell and ConvLSTM branches to model physical dynamics and residual factors, respectively. SimVP (Gao et al., 2022a) adopts a feedforward encoder–temporal module–decoder

architecture for parallel prediction, and TAU (Tan et al., 2023a) combines intra-frame static attention, inter-frame dynamic attention, and differential divergence regularization to capture temporal variations. Other representative architectures include EarthFormer (Gao et al., 2022b), which employs cuboid attention to model high-dimensional spatiotemporal dependencies, and AlphaPre (Lin et al., 2025), which disentangles phase and amplitude changes in the frequency domain to model the positional and intensity evolution of radar echoes, respectively. To address task imbalance in observation-space forecasting, prior work has studied task-aware weighting objectives across precipitation and traffic domains, including B-MSE, B-MAE, and hybrid losses for rare, high-intensity rainfall, as well as peak-sensitive and peak-aware weighted-MSE losses for peak or congested traffic regimes (Shi et al., 2017; Cao et al., 2022; Guo et al., 2021; Fan & He, 2026). Despite the different dynamical mechanisms of these backbones, their prediction supervision, including existing task-weighting schemes, remains primarily in the decoded observation space and does not involve future representation targets constructed by a target encoder or task-differentiated representation matching. Accordingly, we select PhyDNet, SimVP, and TAU as representatives of physics-guided recurrent dynamics, convolutional feedforward dynamics, and attention-modulated dynamics, respectively, and integrate them into a JEPA training framework while preserving their core modules to evaluate task-aware joint-embedding supervision across backbones with different dynamical inductive biases.

### 2.2 Joint-Embedding Prediction and Latent-Space Dynamics

Joint-embedding predictive architectures predict target representations from context representations rather than directly reconstructing target pixels. I-JEPA (Assran et al., 2023) predicts the latent representations of multiple target blocks from a context block within an image, while V-JEPA (Bardes et al., 2024) extends this paradigm to video by predicting masked spatiotemporal regions in representation space; both use masking to partition visible context from prediction targets. For sequential data, the temporal axis provides a natural history–future partition. SPR (Schwarzer et al., 2020) employs a learned transition model to predict multiple future latent states, with an EMA target encoder providing the prediction targets. V-JEPA 2-AC (Assran et al., 2025) instead trains an action-conditioned predictor on top of a frozen video encoder to autoregressively predict future-frame representations for robotic planning.

### 2.3 Task-Aware Supervision in Representation Space

Standard JEPA objectives typically aggregate representation-matching errors uniformly over target units. Recent work has adjusted the focus of representation learning through target construction, prediction ordering, and auxiliary loss weighting. DMT-JEPA (Mo & Yun, 2024) constructs discriminative targets based on feature similarities among neighboring patches. DSeq-JEPA (He et al., 2025) identifies discriminative regions according to the similarity between class and patch tokens and predicts them in order of importance. V-JEPA 2.1 (Mur-Labadia et al., 2026) weights an auxiliary context self-supervision term according to the spatiotemporal distance from each context patch to the nearest masked region. However, these priorities are derived from internal feature relations or predefined geometric distances, and do not address how to use externally specified forecasting events to adaptively infer the task relevance of spatial and channel-wise representation units for each sample. Causal-JEPA (Nam et al., 2026) constructs counterfactual-style prediction tasks through object-level latent masking to encourage the modeling of interaction-dependent dynamics. Perturbation-based attribution methods estimate feature importance by modifying inputs and measuring the resulting model responses, but are primarily designed for post-hoc interpretation (Fong & Vedaldi, 2017; Petsiuk et al., 2018). Unlike methods driven by internal saliency, geometric priors, latent masking, or post-hoc attribution, LIFT-JEPA retains the ground-truth future as the matching target and derives sample-adaptive spatial- and channel-wise weights from target-encoder responses to observation-space task interventions.

## 3 Proposed Approach

LIFT-JEPA augments future joint-embedding prediction with task-intervention-guided latent reweighting. The online encoder and predictor forecast future representations from historical observations, while a shared exponential-moving-average (EMA) target encoder encodes the factual

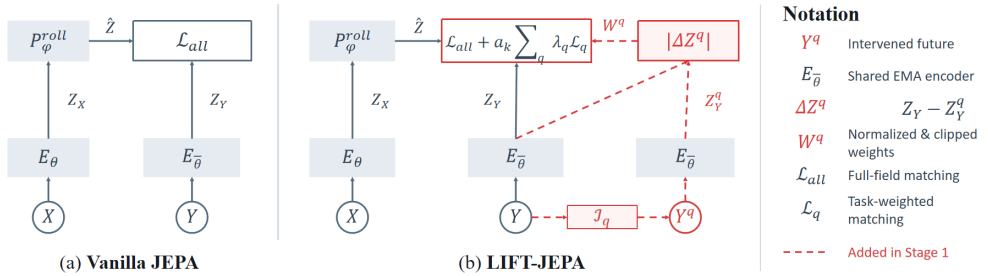



Figure 1: Comparison of Vanilla JEPA and LIFT-JEPA . LIFT-JEPA uses target-encoder responses to task-specific interventions to weight representation matching while retaining the factual target.

and task-intervened futures. Differences between the two target representations provide a proxy for task relevance and yield sample-adaptive spatial–channel weights. These weights define an auxiliary matching loss against the factual target, alongside uniform full-field matching. The intervention branch is used only during representation learning; inference retains the online encoder, predictor, and decoder.

### 3.1 Future Joint-Embedding Prediction

Given a historical observation sequence and its future target,

$$X \in \mathbb{R}^{B\times T\times C\times H\times W}, \qquad Y \in \mathbb{R}^{B\times \tau\times C\times H\times W}, \tag{1}$$

where $B$ is the batch size, $T$ the history length, $\tau$ the prediction horizon, and $(C, H, W)$ the observation dimensions, we decompose a forecasting backbone into an online encoder $E_\theta$, a latent predictor $P_\phi$, and an observation decoder $D_\psi$. Future representations and their targets are

$$Z_X = E_\theta(X), \quad \hat{Z} = P_\phi^{\text{roll}}(Z_X), \quad Z_Y = \text{sg}[E_{\bar{\theta}}(Y)], \tag{2}$$

where $P_\phi^{\text{roll}}$ denotes the backbone's latent forecasting procedure and $\text{sg}[\cdot]$ denotes stop-gradient. Both $\hat{Z}$ and $Z_Y$ have shape $B\times\tau\times C_z\times H_z\times W_z$, with $(C_z, H_z, W_z)$ denoting the latent dimensions. The target encoder is initialized from the online encoder and updated after each Stage-1 optimizer step:

$$\bar{\theta} \leftarrow m\bar{\theta} + (1-m)\theta, \tag{3}$$

where $m \in [0, 1)$ is the EMA momentum.

For factual representation matching, we independently normalize the predicted and target representations within each sample and future time step:

$$\begin{aligned} \mathcal{N}(A) &= \frac{A-\mu(A)}{\sigma(A)+\epsilon}, \\ e &= \rho\Big(\mathcal{N}(\hat{Z}) - \mathcal{N}(Z_Y)\Big), \\ \mathcal{L}_{\text{all}} &= \frac{\text{sum}(e)}{B\tau C_z H_z W_z}, \end{aligned} \tag{4}$$

where $\mu(A)$ and $\sigma(A)$ are the mean and sample standard deviation over $(C_z, H_z, W_z)$ for each sample and time step, broadcast over those dimensions. All operations are elementwise unless a reduction is specified, and sum sums all elements of its input tensor. $\epsilon > 0$ stabilizes the denominators, and $\rho$ is the Smooth-$\ell_1$ function with transition parameter 1. $\mathcal{L}_{\text{all}}$ assigns equal weight to all factual matching errors. We apply this formulation to SimVP (Gao et al., 2022a), TAU (Tan et al., 2023a), and PhyDNet (Guen & Thome, 2020); their encoder, predictor, and decoder instantiations are specified in the experimental setup.

### 3.2 LIFT-JEPA: Task-Intervention-Guided Latent Reweighting

An observation-space task mask identifies relevant pixels or grid cells, but does not directly specify how additional supervision should be allocated across latent channels and locations. LIFT-JEPA estimates this allocation from the target encoder's response to a task-specific intervention.

**Observation-space task intervention.** For a task concept $q$, an intervention operator $\mathcal{I}_q$ modifies selected content in the factual future, optionally using the observed history as a reference:

$$Y^q = \mathcal{I}_q(Y; X), \qquad Z_Y^q = \mathrm{sg}[E_{\bar{\theta}}(Y^q)]\,. \tag{5}$$

The same target encoder, with identical parameters and evaluation mode, encodes $Y$ and $Y^q$. This places their responses in a shared latent coordinate system.

**Teacher-response latent importance.** We measure each latent unit's sensitivity to the intervention by

$$R^q = \frac{|Z_Y - Z_Y^q|}{\sigma(Z_Y) + \epsilon}. \tag{6}$$

This is equivalent to taking the absolute difference after normalizing both target representations with the factual target's statistics. We retain one response per future time step, spatial location, and latent channel, and convert these responses into nonnegative supervision weights:

$$\begin{aligned} \overline{R}^q &= \mathrm{mean}_{\mathrm{lat}}(R^q), \\ W^q &= \mathrm{sg}\left[\mathrm{clip}\left(\frac{R^q}{\overline{R}^q + \epsilon}, 0, \kappa\right)\right], \end{aligned} \tag{7}$$

where $\mathrm{mean}_{\mathrm{lat}}$ averages over $(C_z, H_z, W_z)$ for each sample and time step, with its result broadcast over those dimensions, and $\kappa$ caps the mean-normalized response. Mean normalization makes the weights primarily reflect the relative response pattern within each sample and time step. We use this intervention-conditioned sensitivity as a proxy for task relevance. The weight-generation branch is detached from gradient optimization.

**Interpretation of the intervention response.** For a fixed target encoder and a given pair $(Y, Y^q)$, let $\Delta_q = Y^q - Y$ and view the tensors as vectors. Under local Lipschitz continuity and differentiability almost everywhere along the intervention segment, the fundamental theorem of calculus gives

$$E_{\bar{\theta}}(Y^q) - E_{\bar{\theta}}(Y) = \int_0^1 J_{\bar{\theta}}(Y + \alpha\Delta_q)\Delta_q \,\mathrm{d}\alpha, \tag{8}$$

where $J_{\bar{\theta}}$ is the input Jacobian of the fixed target encoder. The numerator in Eq. (6) is the element-wise magnitude of this finite response. When the perturbation is small and the Jacobian varies little along the segment, Eq. (8) admits the local approximation

$$E_{\bar{\theta}}(Y^q) - E_{\bar{\theta}}(Y) \approx J_{\bar{\theta}}(Y)\Delta_q. \tag{9}$$

LIFT-JEPA uses the finite endpoint difference in Eq. (6), without computing Jacobians or relying on this local approximation. For an affine encoder $E_{\bar{\theta}}(U) = KU + b$, the response is exactly $K\Delta_q$ and depends on the encoder's feature mixing. Thus, pooling the observation mask and broadcasting it across channels does not generally reproduce the encoder-conditioned response weights.

**Task-weighted factual matching.** The weights redistribute the contributions of the factual errors in Eq. (4):

$$\begin{aligned} \mathcal{L}_q &= \frac{\mathrm{sum}(W^q \odot e)}{\mathrm{sum}(W^q) + \epsilon}, \\ \mathcal{L}_{\mathrm{stage1}} &= \mathcal{L}_{\mathrm{all}} + a_s \sum_{q \in \mathcal{Q}} \lambda_q \mathcal{L}_q, \end{aligned} \tag{10}$$

where $\mathcal{Q}$ contains the active task concept, $\lambda_q \geq 0$ controls its auxiliary supervision strength, and $a_s \in [0, 1]$ schedules that strength at Stage-1 epoch $s$. Each auxiliary loss is normalized over the complete minibatch and forecast horizon. A zero response produces a zero weight; if all weights vanish, $\mathcal{L}_q = 0$. The intervened future only determines the weights: both loss terms continue to match $\hat{Z}$ to the factual representation $Z_Y$.

### 3.3 Task-Specific Interventions

We instantiate the same weighting mechanism with an intense-echo intervention for radar nowcasting and a temporal-drift intervention for traffic-flow forecasting.

**Intense-echo intervention for radar nowcasting.** For the single-channel radar observations, we select echoes at or above a reflectivity threshold and replace their values with the no-echo reference:

$$\begin{aligned} M^{\mathrm{int}} &= \mathbb{I}[Y \geq \gamma_{\mathrm{int}}], \\ Y^{\mathrm{int}} &= Y \odot (1 - M^{\mathrm{int}}) + b_0 M^{\mathrm{int}}, \end{aligned} \tag{11}$$

where $\mathbb{I}[\cdot]$ is the indicator function, $\odot$ denotes elementwise multiplication, and both $\gamma_{\mathrm{int}}$ and $b_0$ are expressed on the network input scale. The no-echo value is $b_0 = 0$ under our radar preprocessing; reflectivity thresholds are reported in dBZ in the experiments. For MeteoNet and Shanghai Radar, we use $\mathcal{Q} = \{\mathrm{int}\}$.

**Temporal-drift intervention for traffic-flow forecasting.** For TaxiBJ, we identify task-relevant locations by their cumulative absolute departure from the last observation $X_T$:

$$A = \mathrm{sum}_{\mathrm{time}}(\mathrm{mean}_{\mathrm{chan}}|Y - X_T|), \tag{12}$$

where $\mathrm{mean}_{\mathrm{chan}}$ averages over the $C = 2$ inflow and outflow channels, $\mathrm{sum}_{\mathrm{time}}$ sums over all future steps, and $X_T$ is broadcast across the forecast horizon. Let $\Omega^{(p)}$ contain the $\lceil pHW \rceil$ highest-scoring locations separately for each sample. The intervention mask is

$$M^{\mathrm{drift}} = \mathbb{I}_{\Omega^{(p)}} \odot \mathbb{I}[A > \delta], \tag{13}$$

where $\mathbb{I}_{\Omega^{(p)}}$ is the binary spatial mask of the selected locations. We replace the selected locations with their last observed values across all future steps and channels:

$$Y^{\mathrm{drift}} = \left(1 - M^{\mathrm{drift}}\right) \odot Y + M^{\mathrm{drift}} \odot X_T, \tag{14}$$

with the spatial mask broadcast across future steps and channels. This intervention removes departures from persistence in the selected region; the resulting target-encoder responses determine the spatial–channel loss weights. We use $\mathcal{Q} = \{\mathrm{drift}\}$, $p = 0.20$, and $\delta = 0$. The future-dependent mask is used only for training supervision and is not an input to the online forecasting branch.

### 3.4 Training and Inference

**Representation learning.** In Stage 1, we optimize the online encoder and predictor using Eq. (10), with factual future representations provided by the EMA target encoder. Early in training, representation semantics are still developing, so intervention-derived weights may not reliably reflect task relevance. We therefore initially optimize only $\mathcal{L}_{\mathrm{all}}$ and then linearly increase the auxiliary supervision strength.

**Decoding and fine-tuning.** In Stage 2, we freeze the online encoder and predictor and train the decoder with full-field pixel MSE. We select the checkpoint with the highest validation CSI-M for radar nowcasting or the lowest validation target-region MSE (Target MSE) for TaxiBJ to initialize Stage 3. We then jointly fine-tune all three modules with the same pixel objective. The best Stage-3 checkpoint, selected using the same validation criterion, is used for testing. The target encoder and intervention branch are unused in both stages.

**Inference.** Forecasts depend only on historical observations and model predictions. The intervention and reweighting introduce no additional inference-time inputs, parameters, or computations relative to the same backbone and decoding path. Training and inference details are provided in Appendix B.

## 4 Experiments

### 4.1 Experimental Setup

**Datasets and forecasting protocol.** We evaluate on two precipitation nowcasting datasets, MeteoNet (Larvor et al., 2020) and Shanghai Radar (Chen et al., 2020), and one urban traffic flow dataset, TaxiBJ (Zhang et al., 2017). MeteoNet and Shanghai Radar cover regions of $550\ \mathrm{km} \times 550\ \mathrm{km}$ and $501\ \mathrm{km} \times 501\ \mathrm{km}$, with temporal resolutions of 5 and 6 minutes, respectively. Following DiffCast (Yu et al., 2024), we resize both radar datasets to $128 \times 128$ and predict 20 future

frames from 5 historical frames. TaxiBJ provides inflow and outflow maps of Beijing at 30-minute intervals. Following OpenSTL (Tan et al., 2023b), we use its native $32 \times 32$ resolution and predict 4 future frames from 4 historical frames. For all datasets, we retain the predefined training, validation, and test splits supplied with the data, without repartitioning.

**Backbones and training protocol.** We adopt SimVP (Gao et al., 2022a), TAU (Tan et al., 2023a), and PhyDNet (Guen & Thome, 2020), representing convolutional feed-forward, attention-modulated, and physics-guided recurrent dynamics, respectively. For each dataset–backbone pair, Vanilla JEPA and LIFT-JEPA use identical data processing, model capacity, latent-space rollout, optimizer, training budget, decoder, and model-selection rule. The only difference is that LIFT-JEPA adds the task-aware reweighting term in Stage 1. The intervention thresholds are set to 40 dBZ for Shanghai Radar and 32 dBZ for MeteoNet, matching the corresponding high-threshold evaluation events.

**Evaluation metrics.** Following prior work (Chen et al., 2020; Yu et al., 2024), we evaluate categorical forecast skill using CSI, POD, FAR, and HSS. The evaluation thresholds are $20, 30, 35, 40$ dBZ for Shanghai Radar and $12, 18, 24, 32$ dBZ for MeteoNet. CSI-M is the arithmetic mean of CSI over the corresponding threshold set. CSI$*$L denotes CSI at the lowest threshold, while CSI$*\mathrm{Int}$ denotes CSI at the intervention threshold. They correspond to CSI-20/CSI-40 on Shanghai Radar and CSI-12/CSI-32 on MeteoNet, respectively. POD measures event detection, FAR mea sures false alarms among predicted events, and HSS measures forecast skill relative to chance; all three are averaged over the same threshold set. Ablations additionally report the protocol-specific local metric $\mathrm{CSI}_{16\times16}$. Higher values are better except for FAR. Main results are reported as mean $\pm$ standard deviation over random seeds 0, 1, and 2; ablation tables report means over the same three seeds.

**Implementation details.** All models are implemented in PyTorch 2.3.0 with CUDA 12.1. We adopt a three-stage training protocol, with each stage trained for 80 epochs (240 epochs in total). We use AdamW (Loshchilov & Hutter, 2017) with a batch size of 8 and an initial learning rate of $2 \times 10^{-4}$ throughout training. Within each dataset–backbone setting, JEPA and LIFT-JEPA use identical model capacity, optimization settings, training budget, checkpoint-selection rule, and evaluation procedure. Experiments are conducted on a workstation with an Intel Xeon Gold 6240C CPU at 2.60 GHz, two NVIDIA GeForce RTX 4090 GPUs with 24 GB of memory each, and 64 GB of RAM. Additional training hyperparameters are provided in Appendix A.

### 4.2 Experimental Results

Table 1 presents controlled comparisons across radar nowcasting and urban traffic flow prediction. On the two radar datasets, LIFT-JEPA improves mean CSI-M and $\mathrm{CSI}_{\mathrm{Int}}$ across all six dataset–backbone combinations, with relative gains of $1.30\%$–$2.45\%$ and $3.61\%$–$9.93\%$, respectively. $\mathrm{CSI}_{\mathrm{L}}$ changes by only $-0.55\%$ to $0.97\%$, indicating that the improvements are concentrated near the intervention thresholds. POD and HSS improve by $2.79\%$–$5.78\%$ and $1.04\%$–$1.95\%$, respectively, while FAR increases by $1.48\%$–$5.61\%$, reflecting a detection–false-alarm trade-off. A representative qualitative comparison across the three backbones is provided in Appendix E.

Importantly, Stage-1/2-only evaluations show consistent improvements in CSI-M and $\mathrm{CSI}_{\mathrm{Int}}$ across both radar datasets and two representative backbones, indicating that the task-aligned supervision effect emerges before Stage-3 joint fine-tuning (Appendix F).

For TaxiBJ, Target MSE evaluates the regions selected by Eq. (13), and both methods select checkpoints by minimum validation target-region MSE; global MSE and MAE follow the OpenSTL convention. LIFT-JEPA reduces mean target-region MSE by $5.89\%$, $1.37\%$, and $2.84\%$ on SimVP, PhyDNet, and TAU, respectively, with corresponding global MSE reductions of $4.41\%$, $1.07\%$, and $1.98\%$. The larger relative reductions in target-region MSE are consistent with the intended emphasis on locations undergoing substantial temporal changes. Global MAE increases slightly ($0.06\%$–$1.22\%$), while mean SSIM improves marginally. Together, these results support applying the same latent reweighting mechanism to task priorities defined by either radar intensity or temporal change.

**Operating-point analysis.** We test whether LIFT-JEPA improves POD merely by predicting more events and thereby increasing false alarms. Across all three backbones, Vanilla JEPA exhibits in-

Table 1: Controlled comparisons on Shanghai Radar, MeteoNet, and TaxiBJ. Results are mean ± standard deviation over three random seeds. The better mean within each dataset–backbone pair is shown in bold.

**(a) Radar nowcasting**

| Metric | Shanghai Radar | | MeteoNet | |
|---|---|---|---|---|
| | Vanilla JEPA | LIFT-JEPA | Vanilla JEPA | LIFT-JEPA |
| **SimVP** | | | | |
| CSI-M↑ | $0.4081 \pm 0.0040$ | $\mathbf{0.4134 \pm 0.0082}$ | $0.3870 \pm 0.0042$ | $\mathbf{0.3931 \pm 0.0030}$ |
| $CSI_L$ ↑ | $\mathbf{0.5426 \pm 0.0028}$ | $0.5396 \pm 0.0063$ | $\mathbf{0.5223 \pm 0.0024}$ | $0.5218 \pm 0.0019$ |
| $CSI_{Int}$ ↑ | $0.2561 \pm 0.0070$ | $\mathbf{0.2695 \pm 0.0124}$ | $0.1933 \pm 0.0110$ | $\mathbf{0.2125 \pm 0.0044}$ |
| POD↑ | $0.5193 \pm 0.0119$ | $\mathbf{0.5338 \pm 0.0124}$ | $0.4863 \pm 0.0078$ | $\mathbf{0.5001 \pm 0.0044}$ |
| HSS↑ | $0.5483 \pm 0.0048$ | $\mathbf{0.5540 \pm 0.0091}$ | $0.5237 \pm 0.0058$ | $\mathbf{0.5305 \pm 0.0029}$ |
| FAR↓ | $\mathbf{0.3812 \pm 0.0050}$ | $0.3905 \pm 0.0108$ | $\mathbf{0.3889 \pm 0.0060}$ | $0.3981 \pm 0.0024$ |
| **PhyDNet** | | | | |
| CSI-M↑ | $0.4121 \pm 0.0026$ | $\mathbf{0.4222 \pm 0.0066}$ | $0.4057 \pm 0.0023$ | $\mathbf{0.4110 \pm 0.0025}$ |
| $CSI_L$ ↑ | $0.5468 \pm 0.0013$ | $\mathbf{0.5521 \pm 0.0021}$ | $\mathbf{0.5305 \pm 0.0011}$ | $0.5299 \pm 0.0024$ |
| $CSI_{Int}$ ↑ | $0.2650 \pm 0.0066$ | $\mathbf{0.2756 \pm 0.0119}$ | $0.2213 \pm 0.0037$ | $\mathbf{0.2358 \pm 0.0050}$ |
| POD↑ | $0.5124 \pm 0.0076$ | $\mathbf{0.5336 \pm 0.0158}$ | $0.5167 \pm 0.0028$ | $\mathbf{0.5392 \pm 0.0114}$ |
| HSS↑ | $0.5498 \pm 0.0028$ | $\mathbf{0.5605 \pm 0.0079}$ | $0.5428 \pm 0.0027$ | $\mathbf{0.5495 \pm 0.0036}$ |
| FAR↓ | $\mathbf{0.3677 \pm 0.0076}$ | $0.3734 \pm 0.0051$ | $\mathbf{0.3936 \pm 0.0068}$ | $0.4088 \pm 0.0084$ |
| **TAU** | | | | |
| CSI-M↑ | $0.4160 \pm 0.0057$ | $\mathbf{0.4218 \pm 0.0078}$ | $0.4141 \pm 0.0042$ | $\mathbf{0.4211 \pm 0.0033}$ |
| $CSI_L$ ↑ | $0.5489 \pm 0.0032$ | $\mathbf{0.5529 \pm 0.0063}$ | $0.5402 \pm 0.0029$ | $\mathbf{0.5410 \pm 0.0006}$ |
| $CSI_{Int}$ ↑ | $0.2630 \pm 0.0128$ | $\mathbf{0.2725 \pm 0.0126}$ | $0.2214 \pm 0.0036$ | $\mathbf{0.2413 \pm 0.0100}$ |
| POD↑ | $0.5240 \pm 0.0139$ | $\mathbf{0.5543 \pm 0.0342}$ | $0.5286 \pm 0.0121$ | $\mathbf{0.5434 \pm 0.0091}$ |
| HSS↑ | $0.5547 \pm 0.0071$ | $\mathbf{0.5614 \pm 0.0095}$ | $0.5499 \pm 0.0051$ | $\mathbf{0.5596 \pm 0.0039}$ |
| FAR↓ | $\mathbf{0.3710 \pm 0.0070}$ | $0.3918 \pm 0.0257$ | $\mathbf{0.3857 \pm 0.0108}$ | $0.3914 \pm 0.0054$ |

**(b) Traffic flow prediction on TaxiBJ**

| Backbone | Method | Global MSE↓ | Global MAE↓ | Target MSE ($\times 10^4$)↓ | SSIM↑ |
|---|---|---|---|---|---|
| SimVP | Vanilla JEPA | $0.4967 \pm 0.0024$ | $\mathbf{16.8363 \pm 0.1009}$ | $7.9073 \pm 0.0476$ | $0.970723 \pm 0.000940$ |
| | LIFT-JEPA | $\mathbf{0.4748 \pm 0.0272}$ | $17.0424 \pm 0.2134$ | $\mathbf{7.4417 \pm 0.5132}$ | $\mathbf{0.970937 \pm 0.000186}$ |
| PhyDNet | Vanilla JEPA | $0.4489 \pm 0.0087$ | $\mathbf{16.7815 \pm 0.1280}$ | $6.9109 \pm 0.1845$ | $0.972842 \pm 0.000630$ |
| | LIFT-JEPA | $\mathbf{0.4441 \pm 0.0242}$ | $16.7917 \pm 0.1644$ | $\mathbf{6.8164 \pm 0.5044}$ | $\mathbf{0.972928 \pm 0.001281}$ |
| TAU | Vanilla JEPA | $0.4439 \pm 0.0145$ | $\mathbf{16.0765 \pm 0.1215}$ | $7.0020 \pm 0.2751$ | $0.973283 \pm 0.000678$ |
| | LIFT-JEPA | $\mathbf{0.4351 \pm 0.0148}$ | $16.1478 \pm 0.1786$ | $\mathbf{6.8029 \pm 0.2807}$ | $\mathbf{0.973305 \pm 0.000800}$ |

creasingly severe underforecasting at higher thresholds, whereas LIFT-JEPA moves the mean bias toward, but not above, unity. Although LIFT-JEPA increases both POD and FAR, its operating points reach higher-CSI regions mainly at 30–40 dBZ, indicating that the additional hits are not fully offset by false alarms. Thus, the POD gain is not explained solely by indiscriminate overprediction, but primarily reflects a partial correction of Vanilla JEPA's conservative underforecasting. Detailed frequency-bias curves and performance diagrams are provided in Appendix C.

### 4.3 Ablation Studies

**Protocol.** We conduct ablations on Shanghai Radar using SimVP and report means over random seeds 0, 1, and 2. Within each study, all configurations share the same architecture, training stages, training budget, checkpoint-selection rule, and decoding protocol. The Vanilla JEPA and LIFT-JEPA results correspond to the Shanghai Radar–SimVP results in Table 1, with the same LIFT-JEPA results used in both studies. All percentage changes below are relative to the stated control.

**Compared variants.** Table 2 examines intervention semantics, weight placement, and latent-weight construction. The `uniform` control sets all auxiliary weights to one while retaining the same auxiliary-loss coefficient as LIFT-JEPA. The `random_mask` control replaces the 40-dBZ task-

Table 2: Ablation studies on Shanghai Radar with SimVP. The best and second-best means within each panel are shown in bold and underlined, respectively.

**(a) Mechanism controls**

| Configuration | CSI-M↑ | CSI-20↑ | CSI-40↑ | $CSI_{16\times16}$ ↑ | POD↑ | HSS↑ | FAR↓ |
|---|---|---|---|---|---|---|---|
| Vanilla JEPA | <u>0.4081</u> | **0.5426** | <u>0.2561</u> | 0.5047 | <u>0.5193</u> | <u>0.5483</u> | **0.3812** |
| `uniform` | 0.4061 | 0.5404 | 0.2541 | 0.5067 | 0.5162 | 0.5455 | 0.3823 |
| `random_mask` | 0.4048 | 0.5381 | 0.2543 | 0.5107 | 0.5136 | 0.5442 | <u>0.3822</u> |
| `shuffled_weights` | 0.4061 | <u>0.5405</u> | 0.2558 | <u>0.5205</u> | 0.5178 | 0.5458 | 0.3852 |
| LIFT-JEPA | **0.4134** | 0.5396 | **0.2695** | **0.5320** | **0.5338** | **0.5540** | 0.3905 |

**(b) Latent weight construction**

| Configuration | Latent weighting | CSI-M↑ | CSI-20↑ | CSI-40↑ | $CSI_{16\times16}$ ↑ | POD↑ | HSS↑ | FAR↓ |
|---|---|---|---|---|---|---|---|---|
| `mask_avgpool` | Average-pooled mask | <u>0.4051</u> | <u>0.5338</u> | <u>0.2560</u> | <u>0.5147</u> | <u>0.5207</u> | <u>0.5434</u> | **0.3879** |
| `mask_maxpool` | Max-pooled mask | 0.3976 | 0.5325 | 0.2497 | 0.5070 | 0.5175 | 0.5352 | 0.4074 |
| LIFT-JEPA | Teacher response | **0.4134** | **0.5396** | **0.2695** | **0.5320** | **0.5338** | **0.5540** | <u>0.3905</u> |

aligned mask with a random mask of equal area for each sample, keeping the response calculation and weight placement unchanged. The `shuffled_weights` control retains the task-aligned intervention and weight values but permutes their positions before applying the latent matching loss. For latent-weight construction, we compare average and maximum pooling of the observation-space mask with teacher-response weighting, changing only how the latent weights are constructed.

The `uniform` control remains close to Vanilla JEPA, with CSI-M and CSI-40 decreasing by only $0.49\%$ and $0.78\%$, respectively. In contrast, LIFT-JEPA improves these metrics over the uniform control by $1.80\%$ and $6.06\%$. Thus, adding the same auxiliary term and training schedule without task-dependent weighting does not reproduce the gains of LIFT-JEPA.

The `random_mask` control falls below Vanilla JEPA in CSI-M and CSI-40 by $0.81\%$ and $0.70\%$, respectively. Relative to this control, LIFT-JEPA improves CSI-M and CSI-40 by $2.12\%$ and $5.98\%$, while improving local CSI, POD, and HSS by $1.80\%$–$4.17\%$. These results indicate that introducing task-relevant intervention semantics is more effective than using an arbitrary random intervention.

The `shuffled_weights` control yields a CSI-M of $0.4061 \pm 0.0038$ and a CSI-40 of $0.2558 \pm 0.0065$, returning performance to approximately the Vanilla JEPA level. LIFT-JEPA improves CSI-M and CSI-40 over this control by $1.80\%$ and $5.36\%$, respectively. This result is consistent with the intended role of weight–unit correspondence: preserving the response-weight distribution alone is insufficient when its assignment to latent units is disrupted. Relative to the `random_mask` and `shuffled_weights` controls, LIFT-JEPA increases FAR by $2.17\%$ and $1.38\%$, respectively, consistent with the detection–false-alarm trade-off analyzed in Sec. 4.2.

Compared with average pooling, teacher-response weighting improves CSI-M and CSI-40 by $2.05\%$ and $5.27\%$, respectively, with additional gains in CSI-20, local CSI, POD, and HSS. These improvements are accompanied by a $0.67\%$ increase in FAR. Under this controlled setting, the results support teacher-response weighting as a more effective mapping from observation-space task semantics to latent supervision than the two direct mask-pooling strategies.

## 5 Conclusion

We introduce LIFT-JEPA, which uses target-encoder responses to task-specific interventions to weight the representation-matching loss without additional inference overhead. Experiments on two radar datasets and TaxiBJ show improved task-relevant forecasting across three backbones, while three-seed ablations support the benefits of task-aligned interventions and teacher-response weighting. Future work could explore learning task-relevant latent action representations from multimodal observations to condition latent dynamics. Another direction is to explore intervention-guided post-training of pretrained general-purpose JEPA models to adapt their predictive representations to task-specific priorities.

## AI USE STATEMENT

In this work, we used generative AI tools to assist with translation, manuscript revision, LaTeX formatting, and literature searches. All AI-assisted material included in the paper was reviewed and verified by the authors. The authors take responsibility for the final content of this work, including all text, claims, code, and other artifacts produced with the aid of generative AI.

## REFERENCES

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

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

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

Yuan Cao, Lei Chen, Danchen Zhang, Leiming Ma, and Hongming Shan. Hybrid weighting loss for precipitation nowcasting from radar images. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3738–3742. IEEE, 2022.

Zheng Chang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, Yan Ye, Xiang Xinguang, and Wen Gao. Mau: A motion-aware unit for video prediction and beyond. *Advances in Neural Information Processing Systems*, 34:26950–26962, 2021.

Lei Chen, Yuan Cao, Leiming Ma, and Junping Zhang. A deep learning-based methodology for precipitation nowcasting with radar. *Earth and Space Science*, 7(2):e2019EA000812, 2020.

Xiaolong Fan and Jialin He. Mgttp: A multi-graph transformer model for traffic flow forecasting via bidirectional spatio-temporal interaction. *Chinese Physics B*, 2026.

Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *2017 IEEE international conference on computer vision (ICCV)*, pp. 3449–3457. IEEE, 2017.

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, 2022a.

Zhihan Gao, Xingjian Shi, Hao Wang, Yi Zhu, Yuyang Bernie Wang, Mu Li, and Dit-Yan Yeung. Earthformer: Exploring space-time transformers for earth system forecasting. *Advances in Neural Information Processing Systems*, 35:25390–25403, 2022b.

Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11474–11484, 2020.

Ge Guo, Wei Yuan, Jinyuan Liu, Yisheng Lv, and Wei Liu. Traffic forecasting via dilated temporal convolution with peak-sensitive loss. *IEEE Intelligent Transportation Systems Magazine*, 15(1): 48–57, 2021.

Xiangteng He, Shunsuke Sakai, Shivam Chandhok, Sara Beery, Kun Yuan, Nicolas Padoy, Tatsuhito Hasegawa, and Leonid Sigal. Dseq-jepa: Discriminative sequential joint-embedding predictive architecture. *arXiv preprint arXiv:2511.17354*, 2025.

Gwennaëlle Larvor, Léa Berthomier, Vincent Chabot, Brice Le Pape, Bruno Pradel, and Lior Perez. Meteonet, an open reference weather dataset by meteo-france. 2020. *URL https://www. kaggle. com/datasets/katerpillar/meteonet*, 2020.

Kenghong Lin, Baoquan Zhang, Demin Yu, Wenzhi Feng, Shidong Chen, Feifan Gao, Xutao Li, and Yunming Ye. Alphapre: Amplitude-phase disentanglement model for precipitation nowcasting. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17841–17850. IEEE, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Shentong Mo and Sukmin Yun. Dmt-jepa: Discriminative masked targets for joint-embedding predictive architecture. *arXiv preprint arXiv:2405.17995*, 2024.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-jepa 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026.

Heejeong Nam, Quentin Le Lidec, Lucas Maes, Yann LeCun, and Randall Balestriero. Causal-jepa: Learning world models through object-level latent masking. *arXiv preprint arXiv:2602.11389*, 2026.

Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. *arXiv preprint arXiv:2007.05929*, 2020.

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. *Advances in neural information processing systems*, 28, 2015.

Xingjian Shi, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Deep learning for precipitation nowcasting: A benchmark and a new model. *Advances in neural information processing systems*, 30, 2017.

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, 2023a.

Cheng Tan, Siyuan Li, Zhangyang Gao, Wenfei Guan, Zedong Wang, Zicheng Liu, Lirong Wu, and Stan Z Li. Openstl: A comprehensive benchmark of spatio-temporal predictive learning. *Advances in Neural Information Processing Systems*, 36:69819–69831, 2023b.

Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S Yu. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. *Advances in neural information processing systems*, 30, 2017.

Hao Wu, Yuxuan Liang, Wei Xiong, Zhengyang Zhou, Wei Huang, Shilong Wang, and Kun Wang. Earthfarsser: Versatile spatio-temporal dynamical systems modeling in one model. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 15906–15914, 2024.

Demin Yu, Xutao Li, Yunming Ye, Baoquan Zhang, Chuyao Luo, Kuai Dai, Rui Wang, and Xunlai Chen. Diffcast: A unified framework via residual diffusion for precipitation nowcasting. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27758–27767. IEEE, 2024.

Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Yuchen Zhang, Mingsheng Long, Kaiyuan Chen, Lanxiang Xing, Ronghua Jin, Michael I Jordan, and Jianmin Wang. Skilful nowcasting of extreme precipitation with nowcastnet. *Nature*, 619 (7970):526–532, 2023.

# APPENDIX

## A TRAINING HYPERPARAMETERS

Table 3: Additional hyperparameters and training settings.

| Parameter | Resolved value |
|---|---|
| Auxiliary coefficient $\lambda_q$ | Radar: $\lambda_{\text{int}} = 1.0$; TaxiBJ: $\lambda_{\text{drift}} = 1.0$ |
| Importance clipping $\kappa$ | 10.0 |
| EMA momentum $m$ | 0.996 |
| warm-up and ramp | $S_{\text{warm}} = 8$ epochs and $S_{\text{ramp}} = 8$ epochs |
| Numerical stability $\epsilon$ | $10^{-6}$ |
| Initial learning rate | $2 \times 10^{-4}$ |
| Learning-rate schedule | Cosine decay with a 1000-step warm-up; reset at each stage |
| AdamW | $\beta_1 = 0.90$, $\beta_2 = 0.95$, weight decay $= 0$ |

## B TRAINING AND INFERENCE DETAILS

### B.1 STAGE 1: REPRESENTATION LEARNING

We optimize $E_\theta$ and $P_\phi$ with Eq. (10), matching the predicted representations to the full-horizon factual targets. Gradients propagate through the complete latent rollout without detaching successive predicted blocks. The decoder is frozen, and the target encoder is kept in evaluation mode with no gradient updates. Its parameters follow Eq. (3); any encoder buffers are copied from the online encoder.

**Motivation for delayed auxiliary supervision.** The target encoder's representation semantics can be unstable early in training. Consequently, its responses to task-specific interventions may not yet provide reliable estimates of latent-unit task relevance. Applying task-weighted supervision at this stage could emphasize transient response patterns. We therefore begin with uniform factual representation matching and subsequently introduce the auxiliary loss through a linear ramp.

**Auxiliary-loss schedule.** At Stage-1 epoch $s$, the auxiliary supervision multiplier is

$$a_s = \begin{cases} 0, & s \le S_{\text{warm}}, \\ \min\left\{1, \dfrac{s - S_{\text{warm}}}{\max(S_{\text{ramp}}, 1)}\right\}, & s > S_{\text{warm}}, \end{cases} \tag{15}$$

where $s$ is the one-based Stage-1 epoch index. $S_{\text{warm}}$ denotes the initial number of epochs during which only $\mathcal{L}_{\text{all}}$ is optimized, and $S_{\text{ramp}}$ denotes the subsequent ramp length in epochs. The multiplier $a_s$ is held fixed within each epoch. We use $S_{\text{warm}} = S_{\text{ramp}} = 8$. Disabling the schedule sets $a_s = 1$ throughout Stage 1.

### B.2 STAGE 2: OBSERVATION DECODING

We freeze the online encoder and predictor, keep them in evaluation mode, and train the decoder using full-field mean squared error:

$$\begin{aligned} \hat{Y} &= D_\psi(\hat{Z}; S), \\ \mathcal{L}_{\text{pix}} &= \frac{1}{B\tau CHW} \left\| \hat{Y} - Y \right\|_2^2, \end{aligned} \tag{16}$$

where $S$ denotes the backbone's causal shallow skip features, when present; otherwise $S = \varnothing$. All decoder inputs depend only on historical observations and model predictions.

### B.3 Stage 3: End-to-End Fine-Tuning

An optional third stage jointly fine-tunes $E_\theta$, $P_\phi$, and $D_\psi$ using $\mathcal{L}_{\text{pix}}$, while retaining the same forecasting graph. This stage is used in all reported experiments. In the TAU configuration, predictor BatchNorm running statistics remain fixed, while their affine parameters remain trainable. The EMA target encoder and intervention branch are unused in Stages 2 and 3.

### B.4 Inference

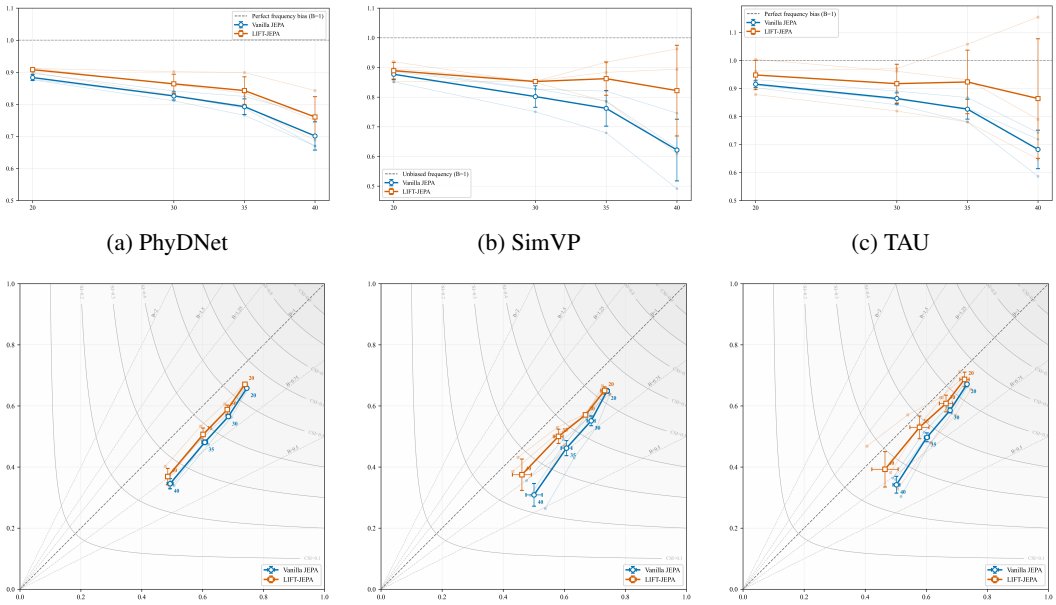
The model forecasts from the observed history using

$$\hat{Y} = D_\psi\big(P_\phi^{\text{roll}}(E_\theta(X)); S\big)\,, \tag{17}$$

with $\hat{Y} \in \mathbb{R}^{B\times\tau\times C\times H\times W}$. The factual future, task mask, and EMA target encoder are not required. Relative to the same forecasting backbone and decoding path, LIFT-JEPA's intervention and reweighting introduce no additional inference-time inputs, parameters, or computations.

## C Operating-Point Analysis

From the per-seed Shanghai Radar results, we compute $\text{SR}_\gamma = 1 - \text{FAR}_\gamma$ and $B_\gamma = \frac{\text{POD}_\gamma}{\text{SR}_\gamma}$ at $\gamma \in \{20, 30, 35, 40\}$ dBZ, and visualize them in Fig. 2.



(a) PhyDNet (b) SimVP (c) TAU

(d) PhyDNet (e) SimVP (f) TAU

Figure 2: Operating-point analysis on Shanghai Radar using PhyDNet, SimVP, and TAU. The top row reports frequency bias, and the bottom row presents the corresponding performance diagrams. Markers and error bars denote the three-seed mean ± one population standard deviation. Connecting lines across reflectivity thresholds are shown only as visual guides.

## D Comparison with Existing Methods

We compare forecasting performance on Shanghai Radar and MeteoNet. Existing-method results are taken from the unified evaluation reported by AlphaPre (Lin et al., 2025). LIFT-JEPA results are means over three random seeds, with standard deviations omitted for compactness. The best and second-best reported means within each dataset are shown in bold and underlined, respectively.

Table 4: External comparisons on Shanghai Radar and MeteoNet. CSI-40 and CSI-32 correspond to the respective intervention thresholds.

| **Shanghai Radar** | | | | | | |
|---|---|---|---|---|---|---|
| Method | CSI-M↑ | CSI-35↑ | CSI-40↑ | HSS↑ | SSIM↑ | MSE↓ |
| ConvGRU (Shi et al., 2017) | 0.3612 | 0.3163 | 0.2062 | 0.4899 | 0.7796 | 33.56 |
| MAU (Chang et al., 2021) | 0.3983 | 0.3621 | 0.2417 | 0.5346 | 0.7195 | 30.40 |
| SimVP (Gao et al., 2022a) | 0.3850 | 0.3549 | 0.2382 | 0.5194 | 0.7795 | 34.40 |
| FourCastNet (Pathak et al., 2022) | 0.3571 | 0.3108 | 0.2073 | 0.4868 | 0.5598 | 32.10 |
| EarthFormer (Gao et al., 2022b) | 0.3503 | 0.3178 | 0.1872 | 0.4844 | 0.7298 | 35.57 |
| PhyDNet (Guen & Thome, 2020) | 0.3654 | 0.3236 | 0.2176 | 0.4957 | 0.7751 | 36.41 |
| EarthFarseer (Wu et al., 2024) | 0.3926 | 0.3608 | 0.2343 | 0.5330 | 0.5405 | 32.68 |
| NowcastNet (Zhang et al., 2023) | 0.3953 | 0.3608 | 0.2450 | 0.5334 | <u>0.7902</u> | 33.56 |
| DiffCast (Yu et al., 2024) | 0.4089 | 0.3740 | 0.2606 | 0.5476 | 0.7879 | 36.35 |
| AlphaPre (Lin et al., 2025) | 0.4178 | 0.3854 | 0.2615 | 0.5534 | **0.7951** | <u>28.02</u> |
| LIFT-JEPA (SimVP) | 0.4134 | 0.3770 | 0.2695 | 0.5540 | 0.7582 | 29.4058 |
| LIFT-JEPA (TAU) | <u>0.4218</u> | <u>0.3885</u> | <u>0.2725</u> | **0.5614** | 0.7419 | 28.1397 |
| LIFT-JEPA (PhyDNet) | **0.4222** | **0.3898** | **0.2756** | <u>0.5605</u> | 0.7780 | **27.8142** |
| **MeteoNet** | | | | | | |
| Method | CSI-M↑ | CSI-24↑ | CSI-32↑ | HSS↑ | SSIM↑ | MSE↓ |
| ConvGRU (Shi et al., 2017) | 0.3401 | 0.2990 | 0.1431 | 0.4667 | 0.7833 | 12.85 |
| MAU (Chang et al., 2021) | 0.3233 | 0.2839 | 0.0997 | 0.4452 | 0.7897 | 12.92 |
| SimVP (Gao et al., 2022a) | 0.3351 | 0.3002 | 0.1130 | 0.4573 | 0.7804 | 13.45 |
| FourCastNet (Pathak et al., 2022) | 0.3027 | 0.2533 | 0.1085 | 0.4216 | 0.6450 | 15.05 |
| EarthFormer (Gao et al., 2022b) | 0.3205 | 0.2884 | 0.1237 | 0.4491 | 0.7772 | 14.43 |
| PhyDNet (Guen & Thome, 2020) | 0.3384 | 0.3194 | 0.1366 | 0.4673 | 0.7823 | 14.48 |
| EarthFarseer (Wu et al., 2024) | 0.3404 | 0.3170 | 0.1372 | 0.4726 | 0.7542 | 14.10 |
| NowcastNet (Zhang et al., 2023) | 0.3427 | 0.3206 | 0.1598 | 0.4751 | 0.7879 | 15.64 |
| DiffCast (Yu et al., 2024) | 0.3512 | 0.3340 | 0.1808 | 0.4846 | 0.7887 | 17.93 |
| AlphaPre (Lin et al., 2025) | 0.3824 | 0.3633 | 0.2002 | 0.5164 | 0.7968 | 12.74 |
| LIFT-JEPA (SimVP) | 0.3931 | 0.3762 | 0.2125 | 0.5305 | 0.7868 | <u>11.0469</u> |
| LIFT-JEPA (TAU) | **0.4211** | **0.4125** | **0.2413** | **0.5596** | <u>0.8224</u> | **10.6403** |
| LIFT-JEPA (PhyDNet) | <u>0.4110</u> | <u>0.4007</u> | <u>0.2358</u> | <u>0.5495</u> | **0.8315** | 11.0914 |

**Shanghai Radar.** As shown in Table 4, LIFT-JEPA with PhyDNet achieves the highest CSI-M, CSI-35, and CSI-40, as well as the lowest MSE on Shanghai Radar, while the TAU-based variant obtains the highest HSS. AlphaPre retains the highest SSIM. Relative to the best results among the existing methods listed, LIFT-JEPA improves CSI-M, CSI-35, CSI-40, and HSS by $1.05\%$, $1.14\%$, $5.39\%$, and $1.45\%$, respectively, while reducing MSE by $0.73\%$.

**MeteoNet.** On MeteoNet, all three LIFT-JEPA variants exceed the listed existing methods in mean CSI-M, CSI-24, CSI-32, and HSS, while achieving lower mean MSE. The TAU-based variant obtains the highest CSI-M, CSI-24, CSI-32, and HSS, with relative improvements of $10.12\%$, $13.54\%$, $20.53\%$, and $8.37\%$ over AlphaPre, respectively, and reduces MSE by $16.48\%$. The PhyDNet-based variant achieves the highest SSIM, exceeding AlphaPre by $4.35\%$. The largest relative gain among the categorical metrics occurs at the intervention threshold, consistent with the emphasis on intense echoes. However, the SimVP-based variant has lower mean SSIM than AlphaPre, indicating that stronger categorical forecast skill does not uniformly translate into better structural similarity.

These external comparisons contextualize absolute forecasting performance under the respective training protocols. The gains attributable specifically to LIFT-JEPA are established by the controlled comparisons with Vanilla JEPA in Table 1, which hold the backbone and training protocol fixed.

## E Qualitative Forecast Visualization

Figure 3 presents a representative forecasting sequence from Shanghai Radar. The upper panel compares the ground-truth future with the SimVP and PhyDNet forecasts, while the lower panel presents the TAU forecasts and enlarged views of the marked core region at $+25$ minutes for all three backbones. In this example, LIFT-JEPA retains stronger echo intensity in the highlighted region and more closely matches the ground truth than Vanilla JEPA across SimVP, PhyDNet, and TAU. This qualitative example complements the aggregate threshold-based results reported in Table 1.

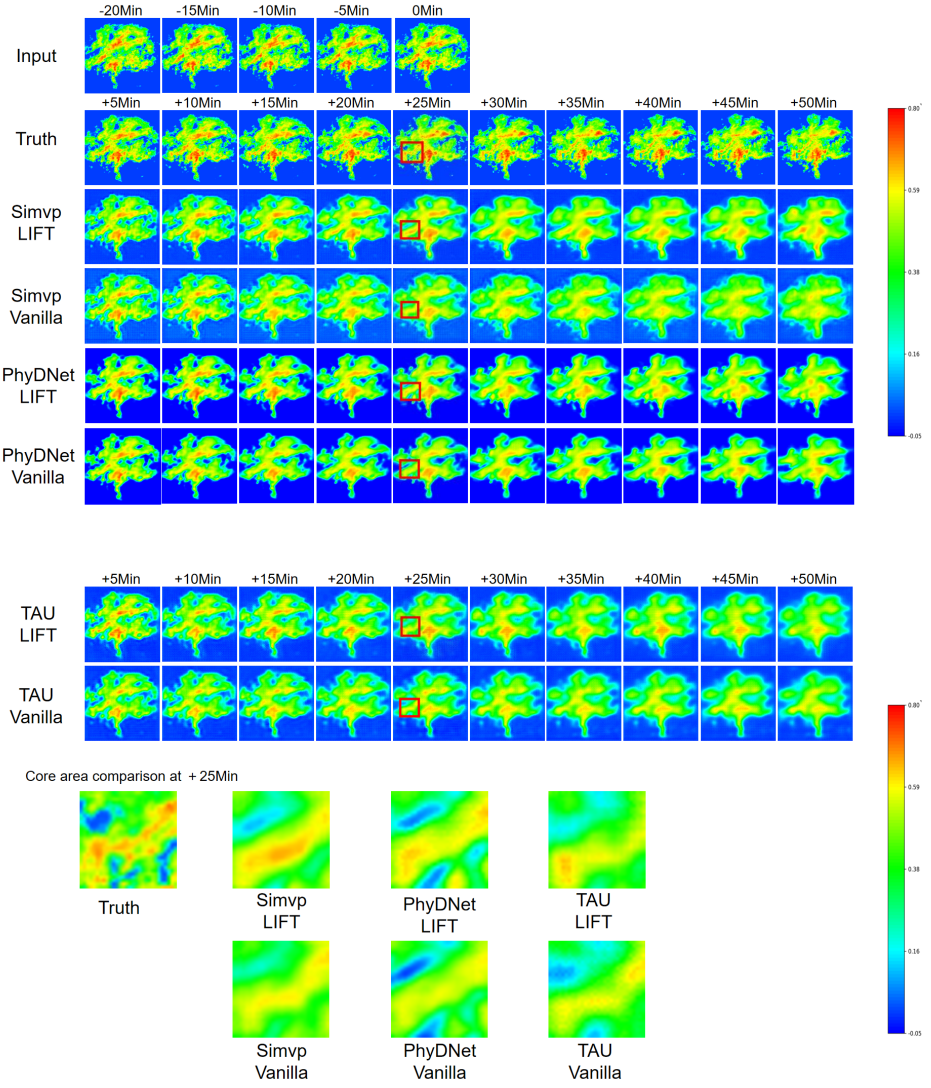



Figure 3: Qualitative comparison on a representative Shanghai Radar sequence. The upper panel shows the input frames, ground-truth future, and forecasts produced by SimVP and PhyDNet. The lower panel shows the TAU forecasts and enlarged views of the red-boxed core region at $+25$ minutes for all three backbones. In the highlighted region, LIFT-JEPA retains stronger echoes that are closer to the ground truth than the corresponding Vanilla JEPA predictions. All panels use the same color scale.

## F TASK-ALIGNED SUPERVISION BEFORE JOINT FINE-TUNING

Table 5: Performance after Stages 1–2 without Stage-3 joint fine-tuning. Results are reported as mean ± standard deviation over three random seeds. Metrics follow Table 1. The better mean within each dataset–backbone pair is shown in bold.

| | Shanghai Radar | | MeteoNet | |
|---|---|---|---|---|
| Metric | Vanilla JEPA | LIFT-JEPA | Vanilla JEPA | LIFT-JEPA |
| *SimVP* | | | | |
| CSI-M↑ | $0.3871 \pm 0.0032$ | $\mathbf{0.3994 \pm 0.0042}$ | $0.3857 \pm 0.0044$ | $\mathbf{0.3884 \pm 0.0022}$ |
| $\mathrm{CSI_L}$ ↑ | $0.5220 \pm 0.0020$ | $\mathbf{0.5265 \pm 0.0020}$ | $0.5214 \pm 0.0029$ | $\mathbf{0.5223 \pm 0.0016}$ |
| $\mathrm{CSI_{Int}}$ ↑ | $0.2456 \pm 0.0034$ | $\mathbf{0.2614 \pm 0.0059}$ | $0.1949 \pm 0.0079$ | $\mathbf{0.2051 \pm 0.0048}$ |
| POD↑ | $0.4934 \pm 0.0085$ | $\mathbf{0.5074 \pm 0.0155}$ | $0.4825 \pm 0.0096$ | $\mathbf{0.4891 \pm 0.0107}$ |
| HSS↑ | $0.5271 \pm 0.0035$ | $\mathbf{0.5390 \pm 0.0048}$ | $0.5229 \pm 0.0058$ | $\mathbf{0.5253 \pm 0.0032}$ |
| FAR↓ | $0.3991 \pm 0.0151$ | $\mathbf{0.3880 \pm 0.0117}$ | $\mathbf{0.3864 \pm 0.0050}$ | $0.3883 \pm 0.0110$ |
| *TAU* | | | | |
| CSI-M↑ | $0.3911 \pm 0.0059$ | $\mathbf{0.3937 \pm 0.0116}$ | $0.3794 \pm 0.0065$ | $\mathbf{0.3852 \pm 0.0059}$ |
| $\mathrm{CSI_L}$ ↑ | $\mathbf{0.5329 \pm 0.0070}$ | $0.5310 \pm 0.0060$ | $\mathbf{0.5271 \pm 0.0027}$ | $0.5267 \pm 0.0050$ |
| $\mathrm{CSI_{Int}}$ ↑ | $0.2388 \pm 0.0035$ | $\mathbf{0.2480 \pm 0.0169}$ | $0.1700 \pm 0.0098$ | $\mathbf{0.1872 \pm 0.0056}$ |
| POD↑ | $\mathbf{0.5000 \pm 0.0122}$ | $0.4991 \pm 0.0327$ | $0.4767 \pm 0.0158$ | $\mathbf{0.4857 \pm 0.0185}$ |
| HSS↑ | $0.5285 \pm 0.0052$ | $\mathbf{0.5311 \pm 0.0155}$ | $0.5107 \pm 0.0081$ | $\mathbf{0.5194 \pm 0.0067}$ |
| FAR↓ | $0.3828 \pm 0.0333$ | $\mathbf{0.3777 \pm 0.0258}$ | $\mathbf{0.3727 \pm 0.0145}$ | $0.3783 \pm 0.0208$ |

Table 5 isolates the effect of LIFT-JEPA before Stage-3 joint fine-tuning. During Stage 2, the online encoder and predictor are frozen and only the decoder is trained under the same pixel objective; the differences between Vanilla JEPA and LIFT-JEPA therefore reflect the representations and predictive dynamics learned in Stage 1. Without Stage 3, LIFT-JEPA improves CSI-M, $\mathrm{CSI_{Int}}$, and HSS in all four dataset–backbone pairs, with relative gains of $0.66\%$–$3.18\%$, $3.85\%$–$10.12\%$, and $0.46\%$–$2.26\%$, respectively. In contrast, changes in $\mathrm{CSI_L}$ are small, showing that the supervision effect is already concentrated near the intervention threshold. Compared with the complete results in Table 1, Stage 3 increases CSI-M for both methods in every evaluated pair, indicating that joint fine-tuning primarily improves overall forecasting performance rather than creating the task-aligned supervision effect.