JEPACAST: PRECIPITATION NOWCASTING IN A DECODABLE AND PREDICTABLE LATENT SPACE

Anonymous authors
Paper under double-blind review

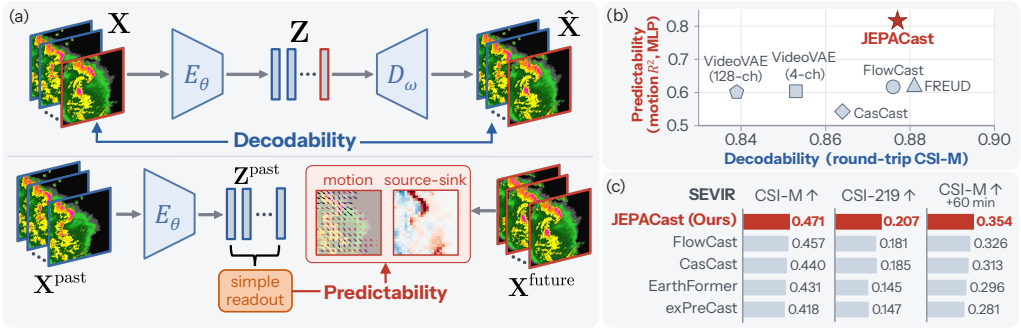

Figure 1: **Overview of JEPACast.** (a) A nowcasting latent should be *decodable*, so that radar fields can be faithfully recovered, and *predictable*, so that future dynamics are exposed to simple readouts. (b) V-JEPA's predictive representation maintains strong decodability while making future physical quantities more predictable by learnable readouts. (c) Forecasting in this space, JEPACast outperforms state-of-the-art models without domain-specific designs.

ABSTRACT

Generative models have become a dominant approach to probabilistic radar-based precipitation nowcasting, yet the latent space where they generate has barely been examined. Most models rely on reconstruction-trained autoencoders, while advances focus on increasingly specialized generators without changing the representation. A simple decomposition of forecast error motivates two fundamental requirements for a forecasting latent space: *decodability* and *predictability*. We find that representations from a V-JEPA encoder trained on radar data remain strongly decodable while achieving higher predictability of future dynamics than widely used reconstruction-based representations. Building on this, we introduce **JEPACast**, which adapts the representation autoencoder paradigm to generative nowcasting by combining the V-JEPA encoder with a latent flow transformer and a ViT decoder. On SEVIR and MeteoNet, JEPACast outperforms various baselines on key forecast metrics, with larger gains for intense precipitation and longer lead times. These results highlight latent representation design as an important direction for generative precipitation nowcasting.

1 INTRODUCTION

Precipitation nowcasting (Shi et al., 2015; Ravuri et al., 2021; Zhang et al., 2023) is critical for short-term weather forecasting, where rapidly evolving convective systems must be predicted at high spatial and temporal resolution. Recent deep generative models (Ravuri et al., 2021; Gao et al., 2023; Foo et al., 2025; Perrone Ribeiro & Faganeli Pucer, 2026) have substantially improved radar-based nowcasting by modeling uncertain futures as distributions rather than single deterministic trajectories. Many generative nowcasters adopt a two-stage design inherited from image and video generation: an autoencoder maps radar observations to a latent space, and a generative model predicts future latents that are decoded back to radar fields (Gao et al., 2023; Leinonen et al., 2023; Foo et al., 2025). Existing approaches have primarily improved the second stage through deterministic

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.

priors (Zhang et al., 2023; Gong et al., 2024), residual or cascaded diffusion (Gong et al., 2024; Yu et al., 2024), motion-aware architectures (Zhang et al., 2023; Wu et al., 2026), and physics-guided objectives (Gao et al., 2023; Wang et al., 2024; Wu et al., 2026). The first stage, however, is still typically trained for reconstruction, preserving information needed to recover radar observations without explicitly targeting future evolution. This leaves a fundamental question underexplored: *what makes a latent representation suitable for precipitation nowcasting?*

A similar shift has emerged in image and video generation, where methods such as RAE (Zheng et al., 2026) move beyond reconstruction-oriented latents and generate in richer representation spaces, demonstrating the importance of latent-space structure for generative modeling. For precipitation nowcasting, however, a useful representation must support both recovery of localized radar patterns and inference of how those patterns evolve.

We argue that a good forecasting latent should satisfy two complementary properties: **decodability** and **predictability**. As illustrated in Figure 1(a), *decodability* measures how faithfully radar fields can be recovered from their latent representations, while *predictability* captures how readily future states can be inferred from past representations. When future-relevant dynamics are poorly organized in latent space, the generator must learn more of this temporal structure itself. Applying these criteria to recent generative nowcasting representations (Gong et al., 2024; Perrone Ribeiro & Faganeli Pucer, 2026; Schusterbauer et al., 2026), we find that reconstruction-trained latents are strongly decodable but expose substantially less future-relevant structure to simple readouts. By contrast, a V-JEPA-style (Mur-Labadia et al., 2026) predictive encoder trained from scratch on SEVIR substantially improves predictability while retaining strong decodability. Figure 1(b) visualizes this separation and suggests that predictive representation learning organizes future-relevant information into a more accessible latent geometry rather than simply preserving more information.

Building on this representation, we introduce **JEPACast**, a generative precipitation nowcasting framework that operates in the predictively learned latent space. JEPACast pairs the frozen V-JEPA encoder with two independently trained modules: a DiT-based flow-matching forecaster (Lipman et al., 2023; Peebles & Xie, 2023) that predicts future JEPA tokens from past tokens, and a ViT decoder that maps them back to radar fields. We adapt RAE principles for generator capacity, noise scheduling, and noise-augmented decoding to spatiotemporal JEPA tokens, yielding a pipeline without nowcasting-specific forecasting modules. As previewed in Figure 1(c), JEPACast translates these representation-level advantages into stronger nowcasting skill on both SEVIR and MeteoNet, with the SEVIR results showing particularly clear gains in high-threshold forecasting (0.207 vs. 0.185) and longer lead-time prediction (0.354 vs. 0.326) over the strongest baseline in each case.

Our contributions are threefold:

- We introduce *decodability* and *predictability* as complementary criteria for nowcasting latents and the diagnostic protocol. We demonstrate that strong reconstruction does not imply high predictability and that V-JEPA retains strong decodability while substantially improving predictability.
- We introduce **JEPACast**, combining a frozen V-JEPA encoder with a flow transformer forecaster and a ViT decoder to model future evolution in a predictively learned latent space without nowcasting-specific forecasting modules.
- On SEVIR and MeteoNet, JEPACast outperforms various deterministic, generative, and hybrid baselines, with clearer advantages toward longer lead times, highlighting latent-space design as complementary to architectural advances.

2 Preliminaries

2.1 Latent Generative Precipitation Nowcasting

Let $\mathbf{x}_t \in \mathbb{R}^{H\times W}$ denote a radar observation at time t. Given a sequence of past observations $\mathbf{X}^{\text{past}} = (\mathbf{x}_1, \ldots, \mathbf{x}_T) \in \mathbb{R}^{T\times H\times W}$, precipitation nowcasting aims to predict the future sequence $\mathbf{X}^{\text{future}} = (\mathbf{x}_{T+1}, \ldots, \mathbf{x}_{T+K}) \in \mathbb{R}^{K\times H\times W}$. Since the future evolution of precipitation is inherently uncertain, generative nowcasting models seek to learn the conditional distribution $p(\mathbf{X}^{\text{future}} \mid \mathbf{X}^{\text{past}})$, from which multiple plausible future trajectories can be sampled (Ravuri et al., 2021; Zhang et al., 2023).

A common approach is to perform this prediction in a learned latent space (Rombach et al., 2022; Gao et al., 2023; Leinonen et al., 2023; Foo et al., 2025). Let E_θ and D_ω denote an encoder and decoder, respectively. The past and future radar sequences are mapped to latent representations $\mathbf{Z}^{\text{past}} = E_\theta(\mathbf{X}^{\text{past}})$ and $\mathbf{Z}^{\text{future}} = E_\theta(\mathbf{X}^{\text{future}})$. A generative forecaster parameterized by ϕ then models $p_\phi(\mathbf{Z}^{\text{future}} \mid \mathbf{Z}^{\text{past}})$, and a sampled future latent $\widehat{\mathbf{Z}}^{\text{future}}$ is mapped back to radar space as $\widehat{\mathbf{X}}^{\text{future}} = D_\omega(\widehat{\mathbf{Z}}^{\text{future}})$. In this formulation, the latent space determines both what information remains available when the forecast is decoded and the representation in which future evolution must be modeled.

Existing latent generative nowcasters (Gao et al., 2023; Leinonen et al., 2023; Foo et al., 2025; Perrone Ribeiro & Faganeli Pucer, 2026) typically obtain this representation from an autoencoder trained primarily for input reconstruction. The first stage therefore encourages E_θ to retain information needed to reproduce the observed radar field, while temporal prediction is left to the downstream generative model. Although this separation has enabled effective latent space forecasting, it does not directly specify which properties of $\mathbf{Z}$ make future states easier to predict. We return to this question in Section 3.

2.2 Joint-Embedding Predictive Architectures

Joint-Embedding Predictive Architectures (JEPAs) (Assran et al., 2023; Bardes et al., 2024) learn representations by predicting missing information in latent space rather than reconstructing the corresponding observations in pixel space. In the V-JEPA family (Bardes et al., 2024; Assran et al., 2025; Mur-Labadia et al., 2026), an input video $\mathbf{X} \in \mathbb{R}^{T\times H\times W\times C}$ is partitioned into non-overlapping spatiotemporal tubelets of size $p_t \times p_h \times p_w$. Each tubelet is projected into a d-dimensional token embedding, yielding $N = (T/p_t) \times (H/p_h) \times (W/p_w)$ tokens. A subset of token locations $\mathcal{M} \subset \{1, \dots, N\}$ is selected as prediction targets, while the remaining locations $\mathcal{C}$ form the visible context. The context encoder E_θ processes the visible token sequence and produces context representations $\mathbf{Z}_\mathcal{C}$. In parallel, the target encoder $E_{\bar\theta}$ processes the complete, unmasked input and produces token-level target representations $\overline{\mathbf{Z}} = (\overline{\mathbf{z}}_1, \dots, \overline{\mathbf{z}}_N)$. The representations $\{\overline{\mathbf{z}}_i\}_{i\in\mathcal{M}}$ at the masked locations serve as prediction targets. A predictor P_ξ takes the context representations together with positional mask tokens indicating the target locations and produces the corresponding predictions $\{\widehat{\mathbf{z}}_i\}_{i\in\mathcal{M}}$. The representation is learned by minimizing

$$\mathcal{L}_{\text{JEPA}} = \frac{1}{|\mathcal{M}|} \sum_{i\in\mathcal{M}} \|\widehat{\mathbf{z}}_i - \text{sg}(\overline{\mathbf{z}}_i)\|_1 \,. \tag{1}$$

Here, sg denotes stop-gradient through the target branch. The target encoder is not directly optimized by this objective. Instead, its parameters are updated as an exponential moving average of the context encoder, $\bar\theta \leftarrow \tau\bar\theta + (1-\tau)\theta$. The predictor is often only used during representation learning, while the learned encoder E_θ is retained for downstream use.

Unlike reconstruction-based objectives, JEPA encourages representations from which missing spatiotemporal content can be inferred from observed context, a property naturally aligned with forecasting. Recent works further suggest that JEPA-based video representations capture information relevant to motion and physical dynamics, including intuitive physics reasoning (Garrido et al., 2025) and prediction of underlying physical parameters of physical systems (Qu et al., 2026). Motivated by these findings, we investigate whether a JEPA representation can serve directly as the generative forecasting space for radar-based precipitation nowcasting.

3 What Makes a Good Forecasting Latent?

In this section, we characterize forecasting latents along two complementary properties, **decodability** and **predictability**. We first motivate these properties from the latent forecasting formulation introduced in Section 2.1, and then examine how they can be measured across representations.

3.1 Decodability and Predictability

Recall that the encoder produces $\mathbf{Z}^{\text{past}} = E_\theta(\mathbf{X}^{\text{past}})$ and $\mathbf{Z}^{\text{future}} = E_\theta(\mathbf{X}^{\text{future}})$, while the generative forecaster models $p_\phi(\mathbf{Z}^{\text{future}} \mid \mathbf{Z}^{\text{past}})$, and a sampled future latent $\widehat{\mathbf{Z}}^{\text{future}}$ is decoded as

$\widehat{\mathbf{X}}^{\text{future}} = D_\omega(\widehat{\mathbf{Z}}^{\text{future}})$. Assume that D_ω is L_D-Lipschitz under latent- and observation-space norms $\|\cdot\|_{\mathcal{Z}}$ and $\|\cdot\|_{\mathcal{X}}$, respectively. Then, for any sampled future latent $\widehat{\mathbf{Z}}^{\text{future}} \sim p_\phi(\cdot \mid \mathbf{Z}^{\text{past}})$, the triangle inequality and the Lipschitz continuity of D_ω give the following pointwise upper bound:

$$\underbrace{\left\|D_\omega(\widehat{\mathbf{Z}}^{\text{future}}) - \mathbf{X}^{\text{future}}\right\|_{\mathcal{X}}}_{\text{forecast error}} \leq \underbrace{\left\|D_\omega(\mathbf{Z}^{\text{future}}) - \mathbf{X}^{\text{future}}\right\|_{\mathcal{X}}}_{\text{reconstruction error}} + \underbrace{L_D \left\|\widehat{\mathbf{Z}}^{\text{future}} - \mathbf{Z}^{\text{future}}\right\|_{\mathcal{Z}}}_{\text{latent prediction error}}. \quad (2)$$

The first term captures the error that remains even when the true future latent is available, while the second bounds the additional error introduced by predicting the future latent from the past. Here, L_D reflects how strongly errors in latent space are amplified after decoding. Although the second term also depends on the forecaster p_ϕ, its difficulty is shaped by how readily the representation exposes information relevant to future evolution.

Decodability. We define decodability as the extent to which a radar field can be recovered from its latent representation. A decodable latent should retain the spatial structure and precipitation intensity needed to reconstruct the observation accurately. This places a basic limit on downstream forecasting, since even a perfect prediction of $\mathbf{Z}^{\text{future}}$ cannot produce an accurate radar field if the latent representation does not preserve the information needed for reconstruction. Since decodability depends on both the encoder and decoder, we evaluate each representation with its corresponding decoder, using the original decoder when available and training one otherwise.

Predictability. We define predictability as the extent to which past latent representations contain future-relevant information that can be readily extracted by a simple predictor, motivated by the notion of usable information under computational constraints (Xu et al., 2020). We therefore assess predictability using low-capacity readouts from frozen past representations, separating properties of the representation from the capacity of a particular generative forecaster. When such information is poorly exposed in $\mathbf{Z}^{\text{past}}$, the generative forecaster $p_\phi(\mathbf{Z}^{\text{future}} \mid \mathbf{Z}^{\text{past}})$ must learn more of the temporal structure needed to model the future latent distribution, increasing the modeling burden on the forecaster. This perspective is also consistent with recent findings that reconstruction quality alone is insufficient for effective latent generation (Yao et al., 2025), as well as with works that augment reconstruction-based latents with predictive objectives to better capture temporal dynamics (Zhao et al., 2026).

Decodability and predictability capture different aspects of a forecasting latent, and neither property alone is sufficient. A collapsed representation may be perfectly predictable while carrying no information needed to recover the radar field. Conversely, a reconstruction-oriented latent may preserve fine-scale information whose future evolution is difficult to infer from the past. Therefore, a useful forecasting latent needs to retain sufficient information for accurate recovery while making future-relevant information accessible to the forecasting model. Additional explanations for these two criteria are in Appendix B.

3.2 Diagnosing Nowcasting Latent Spaces

Diagnostic protocol. We evaluate each frozen representation trained on SEVIR (Veillette et al., 2020) with separate probes for decodability and predictability. For decodability, we decode the full latent sequence back to radar space and report round-trip CSI and SSIM on the reconstructed future frames, without forecasting. For predictability, each readout receives the last 4 context frames (20 minutes) and predicts physical quantities 30 minutes ahead on a 16 km grid using the 5×5 latent neighborhood around each target cell. We consider three properties of future precipitation: **motion**, **intensity**, and **source-sink**. For each cell and lead time, the readout regresses the cell-mean optical flow, the fraction of pixels above VIL 181 in the future frame, and the change in that fraction since the last context frame. These respectively characterize motion, high intensity, and local growth-decay dynamics.

Each model is evaluated using a relative skill score, defined as the fraction of the reference forecast's squared error reduced by the readout. Motion is scored against climatology, while intensity and source-sink are scored against a Lagrangian persistence that transports the current field without growth or decay. We use two simple local readouts, a learned linear probe and a two-layer MLP.

Evaluated latent representations. We evaluate four families of representations under the same diagnostic protocol. First, we consider the frame-wise latent encoders used by existing state-of-

Method	Decodability			Motion		Intensity		Source-sink	
	CSI-M	CSI-219	SSIM	Linear	MLP	Linear	MLP	Linear	MLP
Raw pixels	–	–	–	+0.160	+0.615	+0.123	+0.303	−0.117	+0.247
CasCast	0.864	0.774	0.973	+0.194	+0.533	+0.094	+0.284	−0.207	+0.210
FlowCast	0.876	0.794	0.977	+0.183	+0.619	+0.120	+0.331	−0.125	+0.256
FREUD	**0.881**	**0.796**	0.978	+0.154	+0.617	+0.068	+0.351	−0.189	+0.301
VideoVAE (4-ch)	0.853	0.746	0.973	+0.155	+0.610	+0.148	+0.369	−0.042	+0.330
VideoVAE (128-ch)	0.839	0.653	**0.985**	+0.263	+0.590	**+0.299**	+0.391	+0.204	+0.346
JEPACast	0.877	0.781	0.981	**+0.757**	**+0.806**	**+0.299**	**+0.472**	**+0.250**	**+0.433**

Table 1: **Diagnosis of latents.** Zero indicates no improvement over the corresponding baseline.

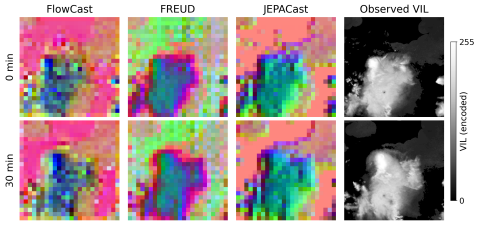

Figure 2: **Latent structure.** PCA-RGB projections at two observations 30 minutes apart, using one basis per model across both observations.

the-art latent generative nowcasting models, including CasCast (Gong et al., 2024), FlowCast (Perrone Ribeiro & Faganeli Pucer, 2026), and FREUD (Schusterbauer et al., 2026). We then evaluate the encoder-decoder pair of JEPACast, the V-JEPA encoder (Mur-Labadia et al., 2026) trained from scratch on SEVIR and a separately trained ViT decoder on its frozen representations. To separate the effect of predictive learning from temporal encoding, we train video VAEs with 4- and 128-channel latents, matching the widths of baselines and JEPACast, respectively.

Results. Table 1 shows that existing latent nowcasting models already achieve strong decodability. Although the encoder of JEPACast is trained without a reconstruction objective, its frozen representation still supports accurate radar reconstruction when paired with a separately trained decoder. The main difference appears in predictability, where JEPACast matches or exceeds every other representation on all three quantities under both readouts. This separation is particularly pronounced under the linear readout, where JEPACast reaches $+0.757$ for motion, substantially above the latents of existing nowcasting models. The gap becomes smaller with the MLP, suggesting that these latents can still contain useful predictive information, but organize it in a form that is less directly accessible to a simple linear mapping. This supports the view that predictive representation learning changes the geometry of the latent space, making future-relevant structure more accessible rather than simply preserving more information. Figure 2 provides a qualitative view of this organization. The projection of the V-JEPA-style encoder more clearly delineates precipitating regions from the background and preserves coherent spatial structure as the precipitation field evolves.

Together, these results suggest that a predictively trained encoder can retain sufficient information for reconstruction while making future-relevant dynamics easier to extract. Full diagnostic details and additional analyses of latent dimensionality, context length, and lead-time dependence are provided in Appendix C.

4 JEPACast

Building on the findings in Section 3, we introduce **JEPACast**, a generative precipitation nowcasting framework that operates in the learned JEPA representation space. As illustrated in Figure 3, JEPACast consists of a JEPA encoder E_θ, a flow-matching transformer forecaster p_ϕ, and a ViT decoder D_ω. The encoder is trained first and then kept fixed, allowing the forecaster and decoder to be optimized independently on the same representation.

Predictive representation learning for radar. Given a radar sequence $\mathbf{X} = (\mathbf{X}^{\text{past}}, \mathbf{X}^{\text{future}}) \in \mathbb{R}^{(T+K)\times H\times W}$, we train the V-JEPA-style encoder from scratch on the full spatiotemporal sequence using the tokenization described in Section 2.2. The resulting tokens are processed with full spatiotemporal attention and factorized 3D-RoPE over time and the two spatial axes (Assran et al., 2025). Building on V-JEPA 2.1 (Mur-Labadia et al., 2026), we augment the masked latent prediction objective in Equation 1 with supervision at visible context locations:

$$\mathcal{L}_{\text{enc}} = \mathcal{L}_{\text{JEPA}} + \mathcal{L}_{\text{ctx}}, \quad \mathcal{L}_{\text{ctx}} = \frac{1}{|\mathcal{C}|}\sum_{i\in\mathcal{C}} \lambda_i \ell(\widehat{\mathbf{z}}_i, \text{sg}(\overline{\mathbf{z}}_i)), \tag{3}$$

where ℓ denotes the smooth-ℓ_1 loss, and $\lambda_i = \lambda_{\text{ctx}}/\sqrt{\max(1, d_{\min}(i, \mathcal{M}))}$ assigns larger weights to visible tokens closer to the masked regions. We further construct prediction targets from nor-

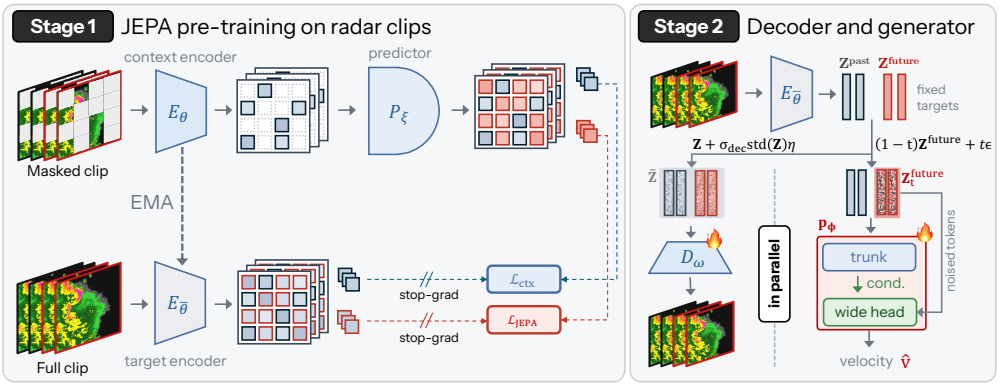

Figure 3: **Pipeline of JEPACast. Stage 1:** A JEPA encoder is trained from scratch on radar clips with masked latent prediction and visible-context supervision. **Stage 2:** A ViT decoder and a flow-matching transformer are trained independently on the fixed representation space of the JEPA EMA target encoder. The decoder reconstructs radar fields from perturbed tokens, while the forecaster predicts future latent velocity from clean past and noised future tokens.

malized features of multiple target encoder layers and train each sample under two spatiotemporal masking views with different scales. After training, we freeze the EMA target encoder $E_{\bar{\theta}}$ and use its representations for both latent forecasting and decoding, denoting them simply as $\mathbf{Z}$ hereafter.

Flow matching transformer. We train a flow-matching transformer p_ϕ to model future evolution in the learned JEPA representation space. Because the encoder uses non-causal spatiotemporal attention, we encode $\mathbf{X}^{\text{past}}$ separately to construct $\mathbf{Z}^{\text{past}}$, while $\mathbf{Z}^{\text{future}}$ is obtained from the future positions of the full-sequence encoding. This prevents future information from leaking into the conditioning representation and ensures that the same context representation is available at training and inference. Following flow matching (Lipman et al., 2023), we construct the noised future representation as $\mathbf{Z}_t^{\text{future}} = (1-t)\mathbf{Z}^{\text{future}} + t\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$ and $t \in [0, 1]$. The forecaster is trained to predict the corresponding velocity $\mathbf{v} = \boldsymbol{\epsilon} - \mathbf{Z}^{\text{future}}$ with

$$\mathcal{L}_{\text{flow}} = \mathbb{E}_{t,\boldsymbol{\epsilon}} \left[\left\| p_\phi \left(\mathbf{Z}^{\text{past}}, \mathbf{Z}_t^{\text{future}}, t \right) - \left(\boldsymbol{\epsilon} - \mathbf{Z}^{\text{future}} \right) \right\|_2^2 \right]. \tag{4}$$

We present the clean past and noised future tokens as a single sequence, with a shared noise level across future frames, following the in-sequence conditioning paradigm of RaMViD (Höppe et al., 2022). With probability p_{uncond}, the context frames are noised together with the future frames to provide unconditional examples for classifier-free guidance (Ho & Salimans, 2021). Following recent representation-space generative models (Porcher et al., 2026; Zheng et al., 2026), we shift noise levels toward stronger corruption (Esser et al., 2024) via $t \leftarrow \alpha t/(1 + (\alpha - 1)t)$.

For the forecaster architecture, we adopt a decoupled trunk-head design based on DDT (Wang et al., 2026a). Following RAE (Zheng et al., 2026), we use a shallow, wide denoising head whose hidden width exceeds the JEPA token dimension, while keeping the global spatiotemporal trunk narrower for computational efficiency. We use full spatiotemporal attention in the forecasting trunk rather than factorized spatiotemporal attention (Ma et al., 2025), allowing direct interactions between any observed and future tokens. We further use sandwich normalization and bounded adaptive residual gates following recent DiT architectures (Zhuo et al., 2024; Wang et al., 2025) to stabilize long-sequence training.

Decoder for reconstructing radar fields. D_ω maps the generated representation $\hat{\mathbf{Z}}^{\text{future}}$ back to the radar prediction $\hat{\mathbf{X}}^{\text{future}}$. We use a frame-wise ViT trained on the full sequence representation $\mathbf{Z} = E_{\bar{\theta}}(\mathbf{X})$ produced by the frozen EMA target encoder, leaving temporal modeling entirely to the forecaster p_ϕ. Following noise-augmented decoding in RAE (Zheng et al., 2026), we perturb the encoder tokens during training as $\tilde{\mathbf{Z}} = \mathbf{Z} + \sigma_{\text{dec}} \operatorname{std}(\mathbf{Z})\boldsymbol{\eta}$, where $\boldsymbol{\eta} \sim \mathcal{N}(0, I)$, exposing the decoder to perturbed representations similar to those produced by the generative forecaster. The decoder is optimized with an ℓ_1 reconstruction objective over all frames:

$$\mathcal{L}_{\text{dec}} = \frac{1}{T+K} \sum_{\tau=1}^{T+K} \left\| D_\omega\left(\tilde{\mathbf{Z}}_\tau\right) - \mathbf{X}_\tau \right\|_1. \tag{5}$$

Table 2: **Comparison on SEVIR.** †Models are trained on 128×128 downsampled SEVIR data and evaluated after upsampling their predictions to the native 384×384 resolution, following prior works. Best and second-best results are shown in **bold** and underline, respectively.

Method	CSI-M↑			CSI-219↑			CRPS↓	SSIM↑	HSS-M↑
	P1	P4	P16	P1	P4	P16			
Deterministic Models									
ConvLSTM (NeurIPS '15)	0.4102	0.4163	0.4475	0.1322	0.1380	0.1734	0.0264	0.7749	0.5232
PredRNN (NeurIPS '17)	0.4045	0.4161	0.4623	0.1331	0.1447	0.1909	0.0271	0.7497	0.5192
PhyDNet (CVPR '20)	0.4198	0.4226	0.4410	0.1362	0.1359	0.1526	0.0253	0.7649	0.5311
SimVP (CVPR '22)	0.4153	0.4226	0.4530	0.1338	0.1394	0.1685	0.0259	0.7772	0.5280
EarthFormer (NeurIPS '22)	0.4310	0.4319	0.4351	0.1448	0.1409	0.1481	0.0251	0.7756	0.5411
exPreCast (ICLR '26)	0.4179	0.4527	0.5427	0.1468	0.1859	0.2910	0.0275	0.7419	0.5430
Hybrid Models									
NowcastNet (Nature '23)	0.4152	0.4452	0.5024	0.1422	0.1874	0.2700	0.0283	0.5696	0.5365
CasCast (ICML '24)	0.4401	0.4640	0.5225	0.1851	0.2127	0.2841	0.0202	0.7797	0.5602
Generative Models									
LDM† (CVPR '22)	0.3465	0.3442	0.3520	0.0671	0.0655	0.0717	0.0208	0.7495	0.4386
PreDiff† (NeurIPS '23)	0.3875	0.3918	0.4157	0.1032	0.1051	0.1213	0.0202	0.7648	0.4914
STLDM† (TMLR '25)	0.4094	0.4332	0.5031	0.1461	0.1730	0.2465	0.0229	0.7802	0.5224
FlowCast (ICLR '26)	0.4573	0.4655	0.4981	0.1806	0.1907	0.2282	**0.0179**	0.8137	0.5753
FREUD (CVPR '26)	0.3831	0.3853	0.3999	0.0868	0.0878	0.0993	0.0182	0.7858	0.4760
JEPACast (Ours)	**0.4715**	**0.5036**	**0.5616**	**0.2071**	**0.2475**	**0.3324**	0.0195	**0.8158**	**0.5942**

Table 3: **Comparison on MeteoNet.**

Method	CSI-M↑			SSIM↑	HSS-M↑
	P1	P4	P16		
Deterministic Models					
ConvGRU (NeurIPS '17)	0.3400	0.3578	0.4473	0.7832	0.4667
PhyDNet (CVPR '20)	0.3384	0.3824	0.4986	0.8022	0.4673
SimVP (CVPR '22)	0.3346	0.3383	0.4143	0.7557	0.4568
EarthFormer (NeurIPS '22)	0.3296	0.3428	0.4333	0.7899	0.4604
Hybrid Models					
DiffCast-ConvGRU (CVPR '24)	0.3512	0.4930	0.7001	0.7761	0.4862
DiffCast-PhyDNet (CVPR '24)	0.3472	0.5066	**0.7200**	0.7788	0.4802
DiffCast-SimVP (CVPR '24)	0.3511	0.5081	0.7155	0.7887	0.4846
Generative Models					
STRPM (CVPR '22)	0.2606	0.4138	0.6882	0.5996	0.3688
MCVD (NeurIPS '22)	0.2336	0.3841	0.6128	0.5414	0.3393
PreDiff (NeurIPS '23)	0.2657	0.3854	0.5692	0.7059	0.3782
JEPACast (Ours)	**0.4263**	**0.5398**	0.6971	**0.8292**	**0.5655**

Table 4: **Ablation on JEPACast Modules**.

Configuration	CSI-M↑	CSI-M-P16↑	CSI-219↑	SSIM↑
Latent Representation				
FlowCast VAE	0.4453	0.5283	0.1864	0.7949
JEPA + ViT Decoder (Ours)	**0.4578**	**0.5450**	**0.1907**	**0.8091**
Flow Transformer Architecture				
Factorized Attn. + DDT Head	0.4514	0.5345	0.1780	0.8073
Full Attn. + No Head	0.4321	0.5149	0.1533	0.7925
Full Attn. + DDT Head (Ours)	**0.4578**	**0.5450**	**0.1907**	**0.8091**
Decoder Objective				
ℓ_1 + LPIPS + GAN	0.4697	**0.5711**	0.2041	0.8132
ℓ_1 (Ours)	**0.4715**	0.5616	**0.2071**	**0.8158**

5 Experiments

5.1 Experimental Setup

Dataset. We evaluate JEPACast on two representative radar-based nowcasting datasets, SEVIR (Veillette et al., 2020) and MeteoNet (Larvor et al., 2020). For SEVIR and MeteoNet, we follow the evaluation protocols of EarthFormer (Gao et al., 2022b) and DiffCast (Yu et al., 2024), respectively. Further details on the dataset are provided in Appendix D.

Implementation details. For SEVIR, the V-JEPA encoder is an 8-block, width-512 ViT with $p_h = p_w = 16, p_t = 1$, and the predictor ViT has 12 blocks of width 256. The flow transformer uses a 12-block, width-512 trunk and a 2-block, width-2048 head, trained with $\alpha = 13$ and $p_{\text{uncond}} = 0.25$. The decoder is a 12-block, width-768 ViT trained with $\sigma_{\text{dec}} = 0.25$. For SEVIR inference, we use 2 Euler steps, $\alpha = 9$, and a linearly increasing CFG weight from 1.25 to 2 across lead times. For MeteoNet, we use the same architectures as for SEVIR except for $p_h = p_w = 8$ and an 8-block, width-512 decoder, with inference using 2 steps, $\alpha = 3$, and a CFG ramp from 1 to 6. Further details are in Appendix E.

Baselines & evaluation metrics. We compare against representative deterministic, hybrid, and generative nowcasting methods on SEVIR and MeteoNet. Details about baselines are described in Appendix F. We evaluate forecasts using Critical Success Index (CSI), Heidke Skill Score (HSS), and Structural Similarity Index (SSIM), reporting CSI-M and HSS-M averaged over the intensity

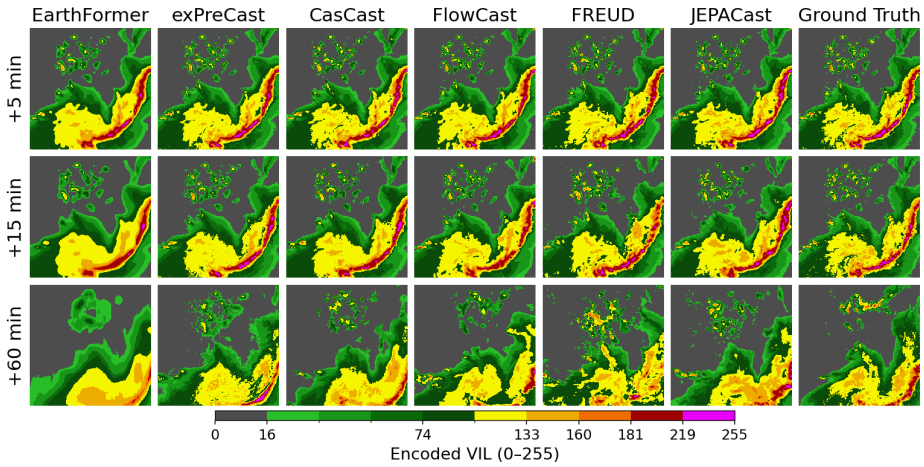

Figure 4: **Qualitative results on SEVIR.** Forecasts from EarthFormer, exPreCast, CasCast, FlowCast, FREUD, and JEPACast are compared with the ground truth at lead times of $+5$, $+15$, and $+60$ minutes. Colors indicate encoded VIL intensity.

thresholds. For SEVIR, we additionally report Continuous Ranked Probability Score (CRPS), CSI-M and CSI-219 at 4×4 and 16×16 spatial pooling scales to account for displacement and assess high-intensity precipitation. On SEVIR, we draw 10 ensemble members per input and compute CSI, HSS, and SSIM on the ensemble mean and CRPS from the member spread, whereas on MeteoNet we evaluate a single sample per input. Metric definitions are in Appendix G.

5.2 Experimental Results

Table 2 reports the precipitation nowcasting results on SEVIR. JEPACast achieves the best reported CSI-M and CSI-219 scores across all pooling scales, together with the best SSIM and HSS-M. At P1, JEPACast improves CSI-M by 3.1% over FlowCast and CSI-219 by 11.9% over CasCast. The advantage is particularly clear for high-intensity precipitation, with CSI-219 gains of 16.4% at P4 and 14.2% at P16 over the strongest baseline at each scale. These results indicate that forecasting in the predictive JEPA representation translates into stronger precipitation forecast skill.

Table 3 shows that the improvement also transfers to MeteoNet. JEPACast improves CSI-M by 21.4% at P1 over the strongest baselines, together with gains of 3.4% in SSIM and 16.3% in HSS-M. Together, these results support the effectiveness of a predictively learned representation as the generative forecasting space across distinct precipitation regimes.

5.3 Analysis

Ablation on JEPACast design choices. Table 4 isolates three design choices under matched settings within each group: the latent representation, the flow transformer architecture, and the decoder objective. The first two groups use a shortened forecaster training schedule. Replacing the JEPA representation with the FlowCast VAE lowers all four metrics, including CSI-M by 0.013 and SSIM by 0.014. Within the forecaster, removing the wide denoising head causes the largest drop (CSI-M -0.026, CSI-219 -0.037), while factorized attention primarily reduces high-intensity skill (CSI-219 0.178 vs. 0.191). For decoding, LPIPS and adversarial terms sharpen the output enough to raise the P16 score but slightly lower every pixel-level metric, so a plain ℓ_1 objective is sufficient for mapping forecasted representations back to radar space. Together, these results support forecasting in the JEPA space with full spatiotemporal attention and a wide head, decoded by an ℓ_1-trained ViT.

Qualitative analysis. Figure 4 compares JEPACast with representative deterministic, hybrid, and generative baselines at +5, +15, and +60 minutes. JEPACast preserves the storm organization and high-intensity precipitation structure more faithfully at longer lead times, consistent with its strong high-threshold forecast skill.

Longer lead-time rollout. Figure 5 extends forecasting to $+180$ minutes by recursively generating 60-minute prediction blocks without additional ground-truth input. Both JEPACast and FlowCast directly reuse generated latents as context for subsequent blocks, decoding them for visualization. In the illustrated example, JEPACast retains more localized precipitation structures while FlowCast produces smoother forecasts, consistent with the predictive JEPA space remaining useful under recursive latent rollout.

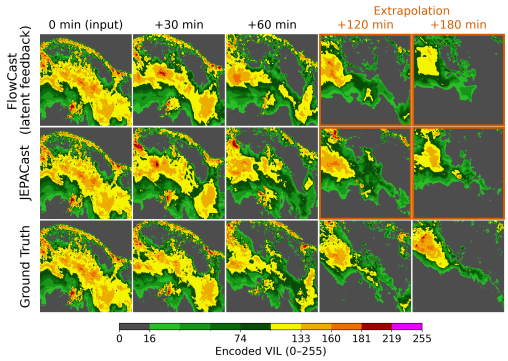

Figure 5: **Longer lead-time rollout on SEVIR.**

6 Related Work

Precipitation nowcasting. Deterministic nowcasting has explored recurrent, convolutional, physics-aware, and transformer-based architectures (Shi et al., 2015; Gao et al., 2022b; Song et al., 2026). Probabilistic and latent generative methods instead model forecast uncertainty using adversarial, diffusion, or flow-based objectives (Ravuri et al., 2021; Gao et al., 2023; Leinonen et al., 2023; Foo et al., 2025; Perrone Ribeiro & Faganeli Pucer, 2026; Schusterbauer et al., 2026). To improve forecast skill, recent approaches further incorporate physical evolution, deterministic-stochastic decomposition, motion guidance, frequency-domain modeling, or precipitation-aware objectives (Zhang et al., 2023; Gong et al., 2024; Wang et al., 2024; Yan et al., 2024; Yu et al., 2024; Lin et al., 2025; Yin et al., 2025; Wu et al., 2026). Yet latent generative nowcasters still typically learn their encoders primarily through reconstruction. JEPACast instead studies how the latent representation itself can preserve radar structure while exposing future-relevant information.

Generative modeling in representation space. Generative modeling has increasingly moved beyond conventional VAE latents toward richer pretrained representations. REPA aligns generative features with pretrained representations, VA-VAE moves this alignment into the autoencoding latent, and RAE directly uses a frozen representation as the generative space (Yu et al., 2025; Yao et al., 2025; Zheng et al., 2026). Subsequent work improves reconstruction from these richer spaces by recovering fine-grained information weakly preserved in deep representations (Chen et al., 2026; Liu et al., 2026; Wang et al., 2026b). Beyond image generation, pretrained representations have also supported predictive modeling for rainfall nowcasting, world modeling, manipulation, video prediction, and stochastic future forecasting (Dourado Filho et al., 2025; Huang et al., 2025; Walker et al., 2026; Zhou et al., 2025; Karypidis et al., 2026). JEPACast brings this perspective to radar nowcasting by characterizing representations through decodability and predictability.

7 Conclusion

We asked what makes a latent space suitable for precipitation nowcasting. A simple decomposition of forecast error yields two requirements, decodability and predictability, and our diagnostic protocol shows that they do not automatically coincide. The reconstruction-trained latents of existing generative nowcasters are highly decodable yet expose little future-relevant structure to simple readouts, whereas a V-JEPA encoder trained on radar data satisfies both, making motion, intensity, and source–sink dynamics more linearly accessible. Our proposed JEPACast turns this decodable and predictable representation into a generative nowcaster by pairing the frozen encoder with a latent flow-matching transformer and a ViT decoder. JEPACast outperforms deterministic, hybrid, and generative baselines on SEVIR and MeteoNet without any nowcasting-specific design, with the largest gains for intense precipitation and longer lead times. These results position latent representation design as a lever complementary to generator design.

AI USE STATEMENT

In this work, we used generative AI tools to edit the manuscript for wording, structure, and readability. We also used them to search and organize related literature and for limited code support. We did not use generative AI tools to generate synthetic data, to formulate or prove mathematical claims, or to clean or reformat datasets. The other disclosure categories were not applicable to this work. All AI-assisted output was reviewed by the authors, references and publication metadata were verified against primary sources, and all reported results and scientific claims were checked by us. We take full responsibility for the final content of this work.

ETHICS STATEMENT

This work uses existing meteorological observations and does not involve human participants or personally identifiable information. The proposed model is intended for research on precipitation nowcasting rather than direct deployment in operational forecasting systems. As with other learned weather forecasting methods, inaccurate predictions, particularly for high-impact precipitation events, could have adverse consequences if used in safety-critical decision-making without appropriate validation and expert oversight.

REPRODUCIBILITY STATEMENT

The main text specifies the representation-learning, forecasting, and decoding objectives, together with the diagnostic protocol used to evaluate decodability and predictability. We provide additional details on the diagnostic protocol and analyses in Appendix C, dataset and nowcasting protocols in Appendix D, implementation and training details in Appendix E, baseline configurations in Appendix F, and evaluation metrics in Appendix G. These appendices provide the experimental settings and configurations required to reproduce the reported results. Training and inference code for the JEPA encoder, the flow transformer, and the decoder, together with the configuration files used for reported SEVIR and MeteoNet experiments, is included in the supplementary material. We will publicly release the code and pretrained checkpoints upon publication.

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 *CVPR*, 2023.

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

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024.

Zheng Chang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, and Wen Gao. STRPM: A spatiotemporal residual predictive model for high-resolution video prediction. In *CVPR*, 2022.

Yitong Chen, Zijie Diao, Junke Wang, Lingyu Kong, Yixuan Ren, Bo He, Yu-Gang Jiang, and Zuxuan Wu. IDEAL: In-depth alignment makes a discrete representation autoencoder. *arXiv preprint arXiv:2606.11096*, 2026.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme Ruiz, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd Van Steenkiste,

Gamaleldin Fathy Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Collier, Alexey A. Gritsenko, Vighnesh Birodkar, Cristina Nader Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetic, Dustin Tran, Thomas Kipf, Mario Lucic, Xiaohua Zhai, Daniel Keysers, Jeremiah J. Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters. In *ICML*, 2023.

Luciano Araujo Dourado Filho, Almir Moreira da Silva Neto, Anthony Miyaguchi, Rodrigo Pereira David, Rodrigo Tripodi Calumby, and Lukáš Picek. DINOv3 as a frozen encoder for CRPS-oriented probabilistic rainfall nowcasting. *arXiv preprint arXiv:2511.10894*, 2025.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

Shi Quan Foo, Chi-Ho Wong, Zhihan Gao, Dit-Yan Yeung, Ka-Hing Wong, and Wai-Kin Wong. STLDM: Spatio-temporal latent diffusion model for precipitation nowcasting. *Transactions on Machine Learning Research*, 2025.

Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z. Li. SimVP: Simpler yet better video prediction. In *CVPR*, 2022a.

Zhihan Gao, Xingjian Shi, Hao Wang, Yi Zhu, Yuyang Wang, Mu Li, and Dit-Yan Yeung. Earthformer: Exploring space-time transformers for earth system forecasting. In *NeurIPS*, 2022b.

Zhihan Gao, Xingjian Shi, Boran Han, Hao Wang, Xiaoyong Jin, Danielle Maddix, Yi Zhu, Mu Li, and Yuyang Wang. PreDiff: Precipitation nowcasting with latent diffusion models. In *NeurIPS*, 2023.

Quentin Garrido, Nicolas Ballas, Mahmoud Assran, Adrien Bardes, Laurent Najman, Michael Rabbat, Emmanuel Dupoux, and Yann LeCun. Intuitive physics understanding emerges from self-supervised pretraining on natural videos. *arXiv preprint arXiv:2502.11831*, 2025.

Junchao Gong, Lei Bai, Peng Ye, Wanghan Xu, Na Liu, Jianhua Dai, Xiaokang Yang, and Wanli Ouyang. CasCast: Skillful high-resolution precipitation nowcasting via cascaded modelling. In *ICML*, 2024.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Tobias Höppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea Dittadi. Diffusion models for video prediction and infilling. *Transactions on Machine Learning Research*, 2022.

Yuhang Huang, Jiazhao Zhang, Shilong Zou, Xinwang Liu, Ruizhen Hu, and Kai Xu. LaDi-WM: A latent diffusion-based world model for predictive manipulation. In *CoRL*, 2025.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

Efstathios Karypidis, Spyros Gidaris, and Nikos Komodakis. Representations before pixels: Semantics-guided hierarchical video prediction. In *ECCV*, 2026.

Gwennaëlle Larvor, Léa Berthomier, Vincent Chabot, Brice Le Pape, Bruno Pradel, and Lior Perez. MeteoNet, an open reference weather dataset by Météo-France, 2020. URL `https://meteonet.umr-cnrm.fr/`.

Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *CVPR*, 2020.

Jussi Leinonen, Ulrich Hamann, Daniele Nerini, Urs Germann, and Gabriele Franch. Latent diffusion models for generative precipitation nowcasting with accurate uncertainty quantification. *arXiv preprint arXiv:2304.12891*, 2023.

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 *CVPR*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023.

Siyu Liu, Chujie Qin, Hubery Yin, Qixin Yan, Zheng-Peng Duan, Chen Li, Jing Lyu, Chun-Le Guo, and Chongyi Li. Improving reconstruction of representation autoencoder. *arXiv preprint arXiv:2602.08620*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Xin Ma, Yaohui Wang, Xinyuan Chen, Gengyun Jia, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *Transactions on Machine Learning Research*, 2025.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Michael 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.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

Bernardo Perrone Ribeiro and Jana Faganeli Pucer. FlowCast: Advancing precipitation nowcasting with conditional flow matching. In *ICLR*, 2026.

François Porcher, Nicolas Carion, Karteek Alahari, and Shizhe Chen. Flow matching in feature space for stochastic world modeling. *arXiv preprint arXiv:2606.29059*, 2026.

Helen Qu, Rudy Morel, Michael McCabe, Alberto Bietti, François Lanusse, Shirley Ho, and Yann LeCun. Representation learning for spatiotemporal physical systems. In *ICLR 2026 Workshop on AI & PDE*, 2026.

Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, et al. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Johannes Schusterbauer, Jannik Wiese, Nick Stracke, Timy Phan, and Björn Ommer. Probabilistic precipitation nowcasting with rectified flow transformers. In *CVPR*, 2026.

Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 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. In *NeurIPS*, 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. In *NeurIPS*, 2017.

Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, pp. 369–386. SPIE, 2019.

Changhoon Song, Teng-Yuan Chang, and Youngjoon Hong. Extreme weather nowcasting via local precipitation pattern prediction. In *ICLR*, 2026.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *NeurIPS*, 2020.

Mark S. Veillette, Siddharth Samsi, and Christopher J. Mattioli. SEVIR: A storm event imagery dataset for deep learning applications in radar and satellite meteorology. In *NeurIPS*, 2020.

Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. MCVD: Masked conditional video diffusion for prediction, generation, and interpolation. In *NeurIPS*, 2022.

Jacob C. Walker, Pedro Vélez, Luisa Polania Cabrera, Guangyao Zhou, Sayna Ebrahimi, Rishabh Kabra, Carl Doersch, Maks Ovsjanikov, João Carreira, and Shiry Ginosar. Frozen forecasting: A unified evaluation. *arXiv preprint arXiv:2507.13942v2*, 2026.

Rui Wang, Jimmy CH Fung, and Alexis KH Lau. Skillful precipitation nowcasting using physical-driven diffusion networks. *Geophys. Res. Lett.*, 51(24):e2024GL110832, 2024.

Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. DDT: Decoupled diffusion transformer. In *CVPR*, 2026a.

Tianhang Wang, Yitong Chen, Wei Song, Zuxuan Wu, Min Li, and Jiaqi Wang. DecQ: Detail-condensing queries for enhanced reconstruction and generation in representation autoencoders. *arXiv preprint arXiv:2605.22777*, 2026b.

Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S. Yu. PredRNN: Recurrent neural networks for predictive learning using spatiotemporal LSTMs. In *NeurIPS*, 2017.

Zhaoqing Wang, Xiaobo Xia, Runnan Chen, Dongdong Yu, Changhu Wang, Mingming Gong, and Tongliang Liu. LaVin-DiT: Large vision diffusion transformer. In *CVPR*, 2025.

Binqing Wu, Weiqi Chen, Shiyu Liu, Zongjiang Shang, Haiou Wang, Liang Sun, and Ling Chen. MoCast: Learning turbulent motions under physical guidance for precipitation nowcasting. In *AAAI*, 2026.

Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. A theory of usable information under computational constraints. In *ICLR*, 2020.

Chiu-Wai Yan, Shi Quan Foo, Van Hoan Trinh, Dit-Yan Yeung, Ka-Hing Wong, and Wai-Kin Wong. Fourier amplitude and correlation loss: Beyond using L2 loss for skillful precipitation nowcasting. In *NeurIPS*, 2024.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *CVPR*, 2025.

Yifang Yin, Shengkai Chen, Yiyao Li, Lu Wang, Ruibing Jin, Wei Cui, and Shili Xiang. SimCast: Enhancing precipitation nowcasting with short-to-long term knowledge distillation. In *ICME*, 2025.

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 *CVPR*, 2024.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *ICLR*, 2025.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

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.

Yian Zhao, Feng Wang, Qiushan Guo, Chang Liu, Xiangyang Ji, Jian Zhang, and Jie Chen. Video generation with predictive latents. *arXiv preprint arXiv:2605.02134*, 2026.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *ICLR*, 2026.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *ICML*, 2025.

Le Zhuo, Ruoyi Du, Han Xiao, Yangguang Li, Dongyang Liu, Rongjie Huang, Wenze Liu, Xiangyang Zhu, Fu-Yun Wang, Zhanyu Ma, Xu Luo, Zehan Wang, Kaipeng Zhang, Lirui Zhao, Si Liu, Xiangyu Yue, Wanli Ouyang, Yu Qiao, Hongsheng Li, and Peng Gao. Lumina-Next: Making lumina-t2x stronger and faster with next-dit. In *NeurIPS*, 2024.

The appendix is organized into the following sections.

- Appendix A: Limitations & Future Works
- Appendix B: Additional Details on Decodability and Predictability
- Appendix C: Diagnostic Protocol and Additional Analysis
- Appendix D: Details on Datasets
- Appendix E: Further Implementation Details
- Appendix F: Details on Baselines
- Appendix G: Evaluation Metrics
- Appendix H: Additional Analysis
- Appendix I: Additional Qualitative Results
- Appendix J: Broader Impacts

A Limitations & Future Works

Limitations. JEPACast relies solely on radar observations and therefore remains limited when future precipitation development is only weakly expressed in the observed history. As illustrated in Figure 15, rapidly developing precipitation can be substantially underpredicted when broad initiation and subsequent growth emerge after the context period. Incorporating additional meteorological observations or large-scale atmospheric context may help address this limitation. Our evaluation is also limited to SEVIR and MeteoNet, and broader validation across geographic regions, radar systems, and precipitation regimes remains necessary. Finally, while our recursive rollouts suggest that the predictively learned representation remains useful beyond the training horizon, systematic evaluation of multi-hour forecasting is left for future work.

Future work. Our current representation learning largely follows the V-JEPA formulation without introducing masking strategies tailored to precipitation dynamics. Designing radar-specific spatial and temporal masking schemes may further improve the predictability of the learned representation. JEPACast also intentionally adopts a relatively simple forecasting pipeline to isolate the effect of the latent space. An important direction is therefore to combine predictive representations with complementary techniques developed for precipitation nowcasting. Another promising direction is to scale predictive representation learning to larger radar archives and study transfer across regions, sensors, and precipitation regimes. Since the encoder is trained self-supervised, such pretraining may provide a reusable forecasting representation beyond a single dataset.

B Additional Details on Decodability and Predictability

Derivation. Let $\|\cdot\|_{\mathcal{Z}}$ and $\|\cdot\|_{\mathcal{X}}$ denote norms on the latent and observation spaces, respectively. We assume that the decoder $D_\omega : \mathcal{Z} \to \mathcal{X}$ is L_D-Lipschitz, such that for any $\mathbf{z}_1, \mathbf{z}_2 \in \mathcal{Z}$,

$$\|D_\omega(\mathbf{z}_1) - D_\omega(\mathbf{z}_2)\|_{\mathcal{X}} \leq L_D \|\mathbf{z}_1 - \mathbf{z}_2\|_{\mathcal{Z}} . \tag{6}$$

Equation 6 makes explicit how a discrepancy measured in latent space is mapped to an upper bound on the corresponding discrepancy after decoding in observation space.

For any sampled future latent $\widehat{\mathbf{Z}}^{\text{future}} \sim p_\phi(\cdot \mid \mathbf{Z}^{\text{past}})$, Equation 2 follows by inserting the decoded true future representation $D_\omega(\mathbf{Z}^{\text{future}})$ between the decoded forecast and the target observation.

Starting from the radar-space forecast error, we obtain

$$\begin{aligned}
&\left\| D_\omega(\widehat{\mathbf{Z}}^{\text{future}}) - \mathbf{X}^{\text{future}} \right\|_{\mathcal{X}} \\
&= \left\| D_\omega(\widehat{\mathbf{Z}}^{\text{future}}) - D_\omega(\mathbf{Z}^{\text{future}}) + D_\omega(\mathbf{Z}^{\text{future}}) - \mathbf{X}^{\text{future}} \right\|_{\mathcal{X}} \\
&\leq \left\| D_\omega(\widehat{\mathbf{Z}}^{\text{future}}) - D_\omega(\mathbf{Z}^{\text{future}}) \right\|_{\mathcal{X}} + \left\| D_\omega(\mathbf{Z}^{\text{future}}) - \mathbf{X}^{\text{future}} \right\|_{\mathcal{X}} \\
&\leq \underbrace{\left\| D_\omega(\mathbf{Z}^{\text{future}}) - \mathbf{X}^{\text{future}} \right\|_{\mathcal{X}}}_{\text{reconstruction error}} + \underbrace{L_D \left\| \widehat{\mathbf{Z}}^{\text{future}} - \mathbf{Z}^{\text{future}} \right\|_{\mathcal{Z}}}_{\text{latent prediction error}}.
\end{aligned} \tag{7}$$

The first inequality follows from the triangle inequality in observation space. The second applies the Lipschitz condition in Equation 6, replacing the decoded latent discrepancy with its latent-space upper bound scaled by L_D. Thus, Equation 2 holds pointwise for each sampled future latent and does not require taking an expectation over p_ϕ.

Interpretation. The two terms highlight different requirements of a forecasting representation. The reconstruction term remains even when the true future latent $\mathbf{Z}^{\text{future}}$ is available, and therefore reflects how faithfully the representation can be mapped back to the target radar field through the decoder. The latent prediction term captures the additional discrepancy introduced when the future representation must instead be predicted from the past. The factor L_D determines how strongly a latent-space prediction error can be amplified when mapped back to observation space. We use this bound only to motivate the separation of decodability and predictability, rather than as a tight characterization of final forecast error.

Predictability diagnostic. The latent prediction term cannot be attributed to the representation alone because it also depends on the forecaster p_ϕ used to produce $\widehat{\mathbf{Z}}^{\text{future}}$. Directly evaluating this term with a particular high-capacity forecaster would therefore conflate representation quality with forecasting-model capacity. We instead probe frozen past representations with low-capacity readouts of future physical quantities, following the notion of usable information under computational constraints (Xu et al., 2020). The resulting predictability score does not directly estimate $\|\widehat{\mathbf{Z}}^{\text{future}} - \mathbf{Z}^{\text{future}}\|_{\mathcal{Z}}$. Rather, it measures how readily future-relevant information can be extracted from the representation by a simple predictor.

C Diagnostic Protocol and Additional Analysis

C.1 Evaluation Setup

We evaluate all frozen representations on SEVIR (Veillette et al., 2020) at the native 384×384 resolution and 5-minute temporal resolution. Each sequence contains 13 context frames followed by 12 future frames. For the predictability analysis, we use the first 25-frame window from each of the 4,053 test events. Using a single window per event avoids temporal overlap between windows assigned to different cross-fitting folds. The decodability probe is scored on the same 4,053 windows, so the two probes share a sample.

To compare representations with different spatial resolutions, we map all latents to a common 24×24 grid of 16 km cells. Representations defined on a 48×48 spatial grid are converted using a 2×2 space-to-depth operation, which preserves their values while matching their spatial support. This results in a per-cell latent width of 512 for the JEPACast encoder and the 128-channel VideoVAE, 256 for raw pixels, and 16 for the remaining reconstruction-based representations. All encoders remain frozen throughout the analysis.

Baselines compared with JEPACast. We evaluate three autoencoders trained on SEVIR using their released weights: CasCast (Gong et al., 2024), FlowCast (Perrone Ribeiro & Faganeli Pucer, 2026), and FREUD (Schusterbauer et al., 2026). To disentangle the effects of the training objective and latent width, we train two VideoVAE variants on the same SEVIR split as JEPACast. Both follow the reconstruction-based autoencoder recipe used by prior baselines, with an ℓ_1 reconstruction loss, KL regularization, and adversarial loss, while operating on 25-frame clips. The 4-channel model

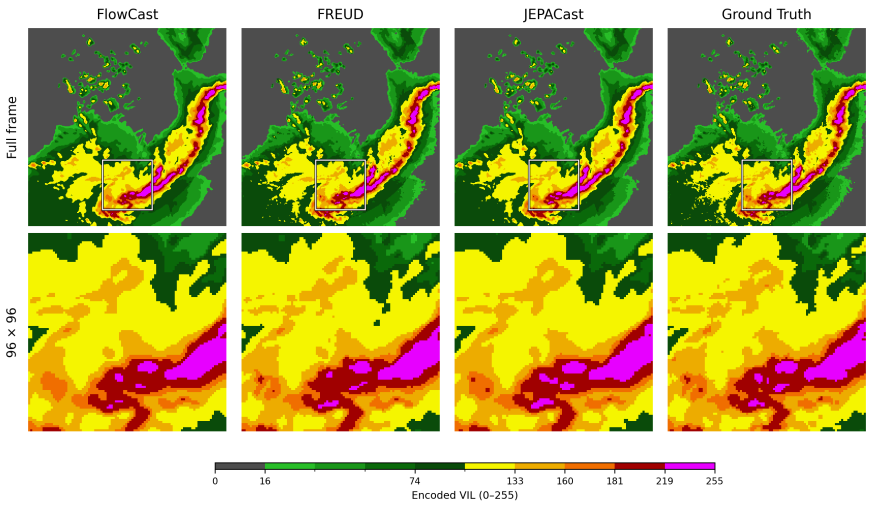

Figure 6: **Qualitative examples of latent decodability.** Ground-truth VIL and reconstructions obtained by encoding and decoding the observations without forecasting. The lower row enlarges the boxed 96×96 region to show fine-scale and high-intensity precipitation structure.

matches the latent dimension of the existing autoencoder baselines, whereas the 128-channel model increases the per-cell latent width to 512 to match JEPACast. We additionally train a width-16 V-JEPA variant under the same predictive objective with a reduced latent width.

C.2 Decodability Probe

Evaluation. We measure decodability by encoding an observed sequence and decoding the resulting representation back to radar space, without any forecasting. The reconstructed future frames are evaluated at the native 384×384 resolution using CSI-M, CSI-219, and SSIM. For existing latent nowcasting models, we use their corresponding decoders. For JEPACast, we first train the V-JEPA-style encoder and train a separate frame-wise ViT decoder on its frozen representations.

Qualitative examples. Figure 6 shows representative round-trip reconstructions without forecasting. The full-frame views illustrate overall precipitation organization, while the enlarged crops highlight localized and high-intensity structure. These examples qualitatively complement the decodability scores reported in Table 1.

C.3 Predictability Probe

The predictability probe measures how readily future-relevant quantities can be extracted from a frozen representation. Each readout uses the 5×5 latent neighborhood surrounding each target cell.

Targets. According to the advection equation with a source term, the precipitation field can be expressed as

$$\partial_t s = -\mathbf{u} \cdot \nabla s + S, \tag{8}$$

where s denotes the precipitation field, $\mathbf{u}$ its motion, and S its local source-sink dynamics. We probe three properties of the future field: motion, intensity, and source-sink. Motion is represented by the cell-averaged optical flow. Intensity is represented by the fraction of pixels within each cell that exceed VIL 181 in the future frame. Source-sink is represented by the change in this fraction relative to the last context frame. All three targets are computed from the ground-truth future frames on the same 24×24 cell grid: optical flow is estimated with the Farnebäck algorithm at the native 384×384 resolution between consecutive frames and then averaged within each 16×16-pixel cell, and the exceedance fraction counts pixels at or above VIL 181 among the 256 pixels of a cell. The motion target at lead l is the instantaneous flow between frames $t + l - 1$ and $t + l$ rather than the accumulated displacement since the last context frame. Unless otherwise stated, intensity and

source-sink are evaluated at a $+30$ minute lead time. Motion and intensity are scored on all 576 cells of every evaluated window. Source-sink is scored only over precipitation cores, cells containing at least one pixel above VIL 160 either at the last context frame or at the evaluated lead, which is 7.6% of all cell-lead pairs, since growth and decay are only defined where there is precipitation to grow or decay. This restriction applies at scoring time only: the readouts are fitted on every cell, without masking or reweighting.

Evaluation. Each target is evaluated using a relative skill score,

$$\text{Skill} = 1 - \frac{\sum(\hat{y} - y)^2}{\sum(y_{\text{ref}} - y)^2}. \tag{9}$$

Both sums run over cells and events before the ratio is taken, so that windows with little precipitation cannot dominate through a near-zero denominator. Motion is scored against climatology: the mean of each target over the fit folds, pooled across all windows and all 576 cells, so the reference is a single constant per lead time that uses neither location nor recent history. Because the reference is a constant predictor fitted on the same folds as the readout, the motion score is a coefficient of determination. Intensity and source-sink are scored relative to Lagrangian persistence, which warps the last context frame along the flow estimated between the two final context frames, scaled by the number of frames to the evaluated lead, using backward sampling with zero padding at the boundary. It translates precipitation but can neither create nor destroy it. A score of zero therefore indicates no improvement over the corresponding reference, while negative values indicate worse performance.

Training readouts. We use two readouts of increasing capacity: a learned linear projection and a two-layer MLP with hidden width 512 and GELU activations. The linear and MLP probes operate on all 576 cells of each window and jointly predict the full set of diagnostic targets. We use five-fold chronological cross-fitting over the 4,053 test events. Each event is evaluated exactly once using a probe fitted on the remaining four folds.

The probes are optimized with AdamW (Loshchilov & Hutter, 2019) and a OneCycle learning-rate schedule (Smith & Topin, 2019) using a batch of eight windows, corresponding to 4,608 cell samples per optimization step. The linear and MLP probes are trained for up to 120 epochs with a learning rate of 10^{-4} and 10^{-3}, respectively. Every representation converges within it, where the selected epoch falls between 17% and 88% of the budget, and validation is flat or declining over the final epochs in all cases. For both probes, the checkpoint with the best performance on an inner validation split is retained.

The distinction between the linear and nonlinear probes is important for interpretation. A low linear score indicates that the target is not readily accessible through a linear mapping, but does not imply that the corresponding information is absent from the representation.

C.4 Additional Diagnostic Analysis

Latent dimensionality analysis. The native per-cell latent width varies from 16 to 512 dimensions across representations. To separate representation quality from the additional capacity afforded by a wider readout input, we fit PCA on SEVIR training windows and project all representations to 16 dimensions per cell before fitting the probes.

Table 5 shows the resulting equal-width comparison. After compression, JEPACast's latents remain the strongest representation for motion, with its MLP score decreasing from $+0.806$ to $+0.731$. A V-JEPA-style encoder trained natively at width 16 reaches $+0.666$, indicating that the advantage is not explained by readout dimensionality alone. In contrast, the advantage on intensity and source-sink is substantially reduced under aggressive compression. Table 6 reports how much of each representation's variance the first 16 principal components capture. At the same nominal width of 512, JEPACast spreads its variance over far more directions than the 128-channel VideoVAE. This suggests that motion information is represented in a comparatively compact subspace, while intensity and growth-decay information rely more strongly on additional latent dimensions.

Figure 7 provides a complementary reconstruction view of the same dimensionality effect. Coarse precipitation organization remains recoverable under strong PCA compression, while additional dimensions progressively restore localized intensity, sharper boundaries, and fine-scale structure. This

Table 5: **Predictability at equal latent width.** Each representation is evaluated at 16 dimensions per cell before fitting the probes. Representations with larger native dimensions are projected to 16 dimensions.

Representation	Native dim.	Motion		Intensity		Source-sink	
		Linear	MLP	Linear	MLP	Linear	MLP
Raw pixels	256	+0.163	+0.655	+0.123	+0.295	−0.107	+0.205
CasCast	16	+0.194	+0.533	+0.094	+0.284	−0.207	+0.210
FlowCast	16	+0.183	+0.619	+0.120	+0.331	−0.125	+0.256
FREUD	16	+0.154	+0.617	+0.068	+0.351	−0.189	+0.301
VideoVAE (4-ch)	16	+0.155	+0.610	+0.148	+0.369	−0.042	**+0.330**
VideoVAE (128-ch)	512	+0.159	+0.644	**+0.183**	**+0.374**	**+0.045**	+0.319
V-JEPA (16-dim)	16	+0.231	+0.666	+0.079	+0.327	−0.251	+0.268
JEPACast	512	**+0.558**	**+0.731**	+0.093	+0.347	−0.239	+0.275

Table 6: **Effective dimensionality of high-dimensional representations.** Fraction of latent variance in the first 16 principal components, using the same basis as Table 5.

Representation	Native dim.	Variance in first 16 PCs
Raw pixels	256	0.99
VideoVAE (128-ch)	512	0.96
JEPACast	512	0.38

is consistent with the quantitative analysis above, showing that the wider JEPA representation carries non-redundant information relevant to localized precipitation details.

Context length. In Table 7, we vary the context available to the readout across 2, 4, and 13 frames and report their results. Across representations, 4 context frames provide the strongest or near-strongest performance for most targets, while increasing the context to all 13 frames does not consistently improve predictability. We therefore use 4 context frames in the main analysis.

Lead-time dependence. We additionally evaluate predictability at $+15$, $+30$, and $+60$ minutes and report in Table 8. JEPACast remains the strongest representation across all three target groups and all reported lead times. Motion predictability decreases with forecast horizon, from $+0.846$ at $+15$ minutes to $+0.712$ at $+60$ minutes, reflecting the increasing difficulty of predicting future motion. Intensity and source-sink show a different trend because their scores are defined relative to Lagrangian persistence. At short lead times, Lagrangian persistence already provides a strong reference, leaving less residual error for the readout to remove. These relative skill scores should therefore be interpreted within a fixed lead time rather than compared directly across forecast horizons.

C.5 Considerations

For decodability probes, decoder capacity and training budget are not controlled across representations because the external models use their original decoders. We therefore interpret this analysis as measuring the decodability of each encoder-decoder pair rather than a decoder-independent property of the encoder.

D Details on Datasets

We evaluate on two public radar nowcasting benchmarks, SEVIR and MeteoNet. We follow the preprocessing and evaluation protocols of EarthFormer (Gao et al., 2022b) and DiffCast (Yu et al., 2024), respectively. The two datasets differ not only in region and resolution but also in how sequences are selected: SEVIR is curated around convective storms, whereas MeteoNet is a continuous archive dominated by light and moderate rain (Song et al., 2026).

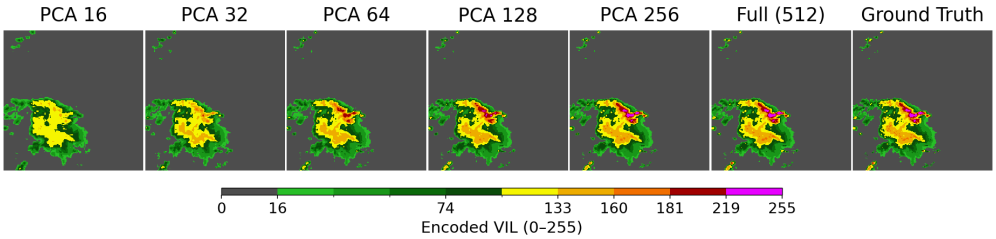

Figure 7: **Decodability under PCA compression.** Reconstructions from JEPA representations projected onto increasing numbers of principal components are compared with the full 512-dimensional representation and the ground truth. Colors indicate encoded VIL intensity.

Table 7: **Effect of context length on predictability.** Results use the 30-epoch trained MLP readout at a +30 minute lead time. Values correspond to 2 / 4 / 13 context frames.

Representation	Motion	Intensity	Source-sink
Raw pixels	+0.519 / +0.539 / +0.520	+0.366 / +0.350 / +0.310	+0.312 / +0.273 / +0.213
CasCast	+0.519 / +0.543 / +0.523	+0.132 / +0.282 / +0.260	+0.229 / +0.213 / +0.172
FlowCast	+0.612 / +0.617 / +0.584	+0.363 / +0.372 / +0.345	+0.300 / +0.302 / +0.265
FREUD	+0.585 / +0.623 / +0.563	+0.355 / +0.361 / +0.331	+0.309 / +0.304 / +0.264
VideoVAE (4-ch)	+0.588 / +0.604 / +0.566	+0.403 / +0.409 / +0.376	+0.376 / +0.365 / +0.313
VideoVAE (128-ch)	+0.572 / +0.603 / +0.562	+0.396 / +0.411 / +0.399	+0.363 / +0.373 / +0.346
JEPACast	**+0.803 / +0.821 / +0.819**	**+0.469 / +0.485 / +0.465**	**+0.439 / +0.451 / +0.417**
V-JEPA (16-dim)	+0.651 / +0.680 / +0.637	+0.320 / +0.348 / +0.320	+0.279 / +0.285 / +0.239

SEVIR. The Storm EVent ImageRy (SEVIR) dataset (Veillette et al., 2020) contains radar, satellite, and lightning observations over the continental United States from 2017 to 2019 and is distributed without restrictions on use. We use the NEXRAD Vertically Integrated Liquid (VIL) modality, provided on a 384×384 grid at 1 km spatial resolution and a 5-minute cadence. Each event spans four hours and contains 49 frames and is divided into three 25-frame windows starting at frames 0, 12, and 24. The task is to predict 12 future frames (60 min) from 13 observed frames (65 min). Following prior works (Gao et al., 2022b; Gong et al., 2024; Schusterbauer et al., 2026), we use the same evaluation protocol from EarthFormer. CSI and HSS are evaluated at the thresholds $\mathcal{T}_{\text{SEVIR}} = \{16, 74, 133, 160, 181, 219\}$. SEVIR includes both randomly sampled events and events associated with severe weather reports, with random sampling biased toward high-VIL regions to reduce precipitation-free scenes.

MeteoNet. MeteoNet (Larvor et al., 2020) is an open meteorological dataset released by Météo-France under the Etalab Open Licence 2.0, containing rain-radar, satellite, ground-station, and weather-model observations over France from 2016 to 2018. Following DiffCast (Yu et al., 2024), we use the rain-radar data from the north-western quarter of France at a 6-minute cadence. Frames are resized from the native 565×784 grid at 0.01° resolution ($\approx$1 km) to 128×128, and reflectivity is clipped to $[0, 70]$ dBZ. We use the training and test windows released by DiffCast (Yu et al., 2024). These 25-frame windows are extracted from the continuous radar archive by filtering out dry periods, with consecutive windows within a rainy period overlapping by 10 frames. The task is to predict 20 future frames (120 min) from 5 observed frames (30 min). CSI and HSS are evaluated at $\mathcal{T}_{\text{MeteoNet}} = \{12, 18, 24, 32\}$ dBZ and averaged over thresholds and lead times following DiffCast. Unlike the storm-focused SEVIR benchmark, MeteoNet is dominated by lighter precipitation, so 20.9% of pixels in the training windows are non-zero, 6.8% exceed 12 dBZ, and 0.35% exceed 32 dBZ.

E Further Implementation Details

Table 9 summarizes the main SEVIR configuration. We provide below the implementation details not specified in the main text, followed by the changes used for MeteoNet and the ablations in Table 4.

Table 8: **Predictability across various forecast lead times.** Results use the MLP readout and four context frames.

	Motion			Intensity			Source-sink		
Representation	+15m	+30m	+60m	+15m	+30m	+60m	+15m	+30m	+60m
Raw pixels	+0.648	+0.615	+0.527	−0.041	+0.303	+0.376	+0.045	+0.247	+0.259
CasCast	+0.562	+0.533	+0.453	−0.099	+0.284	+0.387	−0.001	+0.210	+0.235
FlowCast	+0.653	+0.619	+0.533	−0.002	+0.331	+0.381	+0.079	+0.256	+0.245
FREUD	+0.653	+0.617	+0.531	+0.095	+0.351	+0.383	+0.133	+0.301	+0.288
VideoVAE (4-ch)	+0.646	+0.610	+0.525	+0.150	+0.369	+0.390	+0.174	+0.330	+0.307
VideoVAE (128-ch)	+0.617	+0.590	+0.503	+0.219	+0.391	+0.400	+0.193	+0.346	+0.328
JEPACast	**+0.846**	**+0.806**	**+0.712**	**+0.351**	**+0.472**	**+0.443**	**+0.301**	**+0.433**	**+0.385**
V-JEPA (16-dim)	+0.702	+0.666	+0.578	+0.060	+0.327	+0.383	+0.104	+0.268	+0.274

Table 9: **Hyperparameters of JEPACast on SEVIR.**

	Encoder E_θ / Predictor P_ξ	**Flow Transformer p_ϕ**	**Decoder D_ω**
Architecture			
Input	25 frames at 384^2	25 frames × 576 tokens	576 tokens per frame
Blocks × width	8×512 / 12×256	trunk 12×512, head 2×2048	12×768
Attention	global, 3D RoPE	trunk: global, 3D RoPE head: frame-wise, 2D RoPE	frame-wise, learned positions
MLP	GELU, ratio 4	SwiGLU, hidden 1536 / 5632	GELU, ratio 4
Normalization	pre-LN	pre-RMSNorm gain-free post-RMSNorm tanh gates, QK-norm	pre-LN
Parameters	25.4M / 10.1M	225.4M	86.1M
Optimization			
Objective	$\mathcal{L}_{\text{enc}}$ with smooth-ℓ_1 (β=1), $\lambda_{\text{ctx}} = 0.5$	velocity MSE $\mathcal{L}_{\text{flow}}$	ℓ_1
Optimizer	AdamW	AdamW	Adam
(β, ϵ)	$(0.9, 0.95), 10^{-6}$	$(0.9, 0.95), 10^{-8}$	$(0.9, 0.999), 10^{-8}$
Weight decay	0.03 (weight matrices only)	0	0
Learning rate	3×10^{-4}, constant	10^{-4}, constant	2×10^{-4}, constant
Warm-up	700 steps from $0.1\times$	2,500 steps from 0	–
Grad. clipping	3.0	1.0	–
Training Setup			
Total batch size	512	128	128
Training length	15k update steps	45k update steps	20k update steps
GPU	8 B200	8 H100 SXM	8 H100 SXM
Training time	∼33h	∼32h	∼6h
EMA	target encoder, $\tau = 0.999$	weights, 0.9999	–
Noise/masking	two masking views	$\alpha = 13$, $p_{\text{uncond}} = 0.25$	$\sigma_{\text{dec}} = 0.25$
Precision	bf16	bf16, TF32	bf16

E.1 JEPA Encoder

Inputs are standardized using SEVIR training statistics and patchified with a 3D convolution of kernel and stride $(1, 16, 16)$, giving 24×24 tokens per frame. The encoder uses factorized 3D-RoPE over time and the two spatial axes, with no absolute positional embeddings. The predictor maps visible tokens to width 256 and predicts 2048-dimensional targets formed by concatenating normalized features from encoder blocks 2, 4, 6, and 8. Each clip is trained with two spatiotemporal masking views: one uses 8 spatial blocks covering 15% of the grid each and the other uses 2 blocks covering 70% each. The blocks span all 25 frames, and at least 64 tokens remain visible.

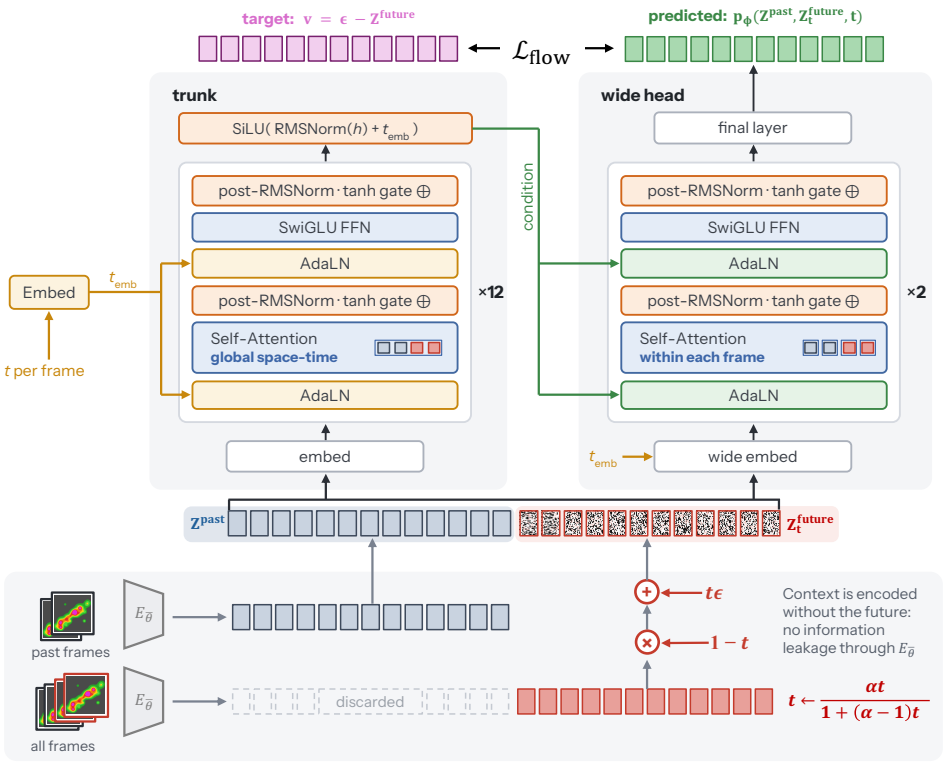

Figure 8: **Architecture of Our Flow Transformer.**

We use the objective in Equation 3. Both the masked-token and visible-context terms use smooth-ℓ_1 loss with $\beta = 1$. For a visible token i, the context weight is $\lambda_i = \lambda_{\text{ctx}}/\sqrt{\max(1, d_{\min}(i, \mathcal{M}))}$ with $\lambda_{\text{ctx}} = 0.5$. Here, $d_{\min}(i, \mathcal{M})$ is the minimum Euclidean distance from token i to the masked set $\mathcal{M}$ on the 3D token grid. The two masking-view losses are averaged, and the target encoder is updated by EMA with $\tau = 0.999$.

E.2 Flow Transformer

For each training window, the frozen target encoder is run once on all 25 frames to obtain the 12 future target latents and separately on the 13 context frames to obtain $\mathbf{Z}^{\text{past}}$. This separate context encoding prevents future information from entering the conditioning representation through the non-causal encoder. Tokens are standardized per channel using statistics computed from the training set.

We draw $n \sim \mathcal{N}(0, 1)$, set $t = \text{sigmoid}(n)$, and shift the noise level as $t \leftarrow \alpha t/[1 + (\alpha - 1)t]$ with $\alpha = 13$. Conditional examples keep the context clean and use the same t for all future frames. With probability $p_{\text{uncond}} = 0.25$, the context is noised at the same level to provide unconditional examples for classifier-free guidance. Future latents are perturbed as $\mathbf{Z}_t^{\text{future}} = (1 - t)\mathbf{Z}^{\text{future}} + t\boldsymbol{\epsilon}$, with $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$, and the model predicts the velocity $\boldsymbol{\epsilon} - \mathbf{Z}^{\text{future}}$.

Figure 8 gives the detailed forecaster architecture. The 12-block width-512 trunk applies full spatiotemporal attention with 3D-RoPE. Its output conditions a 2-block width-2048 denoising head whose attention is restricted within each frame and uses 2D-RoPE. Noise levels are embedded with Gaussian Fourier features (Tancik et al., 2020) followed by a two-layer SiLU MLP. Both trunk and head use AdaLN conditioning, SwiGLU feed-forward layers (Shazeer, 2020), QK normalization (Dehghani et al., 2023), gain-free post-RMSNorm, and bounded tanh residual gates. The final head maps back to the 512-dimensional JEPA token space.

E.3 DECODER

The decoder operates frame-wise on frozen JEPA tokens. A linear projection maps each 512-dimensional token to width 768, after which 12 pre-LN ViT blocks with learned spatial positions produce 16×16 output patches. During training, we apply the noise augmentation, $\widetilde{\mathbf{Z}} = \mathbf{Z} + \sigma_{\mathrm{dec}}\,\mathrm{std}(\mathbf{Z})\boldsymbol{\eta}$, where $\boldsymbol{\eta} \sim \mathcal{N}(0, I)$, $\mathrm{std}(\mathbf{Z})$ is the scalar standard deviation of token batch, and $\sigma_{\mathrm{dec}} = 0.25$. The decoder is trained with an ℓ_1 loss over all frames.

E.4 SAMPLING AND SCORING

At inference, future slots are initialized with Gaussian noise and integrated from $t = 1$ to $t = 0$ using Euler steps on the shifted grid

$$t_i = \frac{\alpha(i/S)}{1 + (\alpha - 1)i/S}, \qquad i = S, \ldots, 0.$$

The update is $\mathbf{Z}_{i-1} = \mathbf{Z}_i - (t_i - t_{i-1})\hat{\mathbf{v}}(\mathbf{Z}_i, t_i)$, with the clean context reinserted after each step. For SEVIR, we use $S = 2$ and $\alpha = 9$, corresponding to evaluations at $t = 1$ and $t = 0.9$ before the final step to zero. Classifier-free guidance uses $\hat{\mathbf{v}} = \hat{\mathbf{v}}_u + w_k(\hat{\mathbf{v}}_c - \hat{\mathbf{v}}_u)$, where w_k increases linearly from 1.25 to 2.0 over the 12 lead times. We generate 10 ensemble members per input.

E.5 METEONET

We retain the SEVIR architecture and optimization settings except for the changes below. The encoder uses 8×8 patches on 128×128 frames while keeping the same transformer depths and widths as on SEVIR, so it produces 16×16 tokens per frame. The encoder is trained for 10k steps with batch size 512. Because MeteoNet is dominated by dry pixels, nonzero-echo pixels (at least $0.5/255$) receive weight 4 in the decoder ℓ_1 loss. The decoder is an 8-block, width-512 ViT, which is trained for 20k steps with batch size 128 and $\sigma_{\mathrm{dec}} = 0.25$. The flow transformer keeps the SEVIR architecture and operates on 5 context and 20 future frames, using a batch size of 256 for 20k steps. The complete MeteoNet training pipeline takes approximately 15.5 hours on 8 NVIDIA H100 GPUs. Inference uses two Euler steps, $\alpha = 3$, and a CFG weight increasing linearly from 1 to 6 across the 20 lead times.

E.6 ABLATION SETTINGS

All ablations in Table 4 use SEVIR. Latent and forecaster ablations are trained for 10k steps with the flow-transformer optimization and sampling settings above.

Latent and forecaster ablations. For the FlowCast-VAE ablation, we use the released $48 \times 48 \times 4$ latent (Perrone Ribeiro & Faganeli Pucer, 2026) and apply a 2×2 space-to-depth rearrangement to obtain the same 24×24 token grid as JEPACast. Only the flow transformer's input and output projections are changed, and predictions are decoded with the released FlowCast decoder. The factorized-attention variant replaces each global trunk attention layer with frame-wise spatial attention followed by temporal attention at each spatial location. The no-head variant removes the width-2048 denoising head and predicts velocity directly from the width-512 trunk.

Decoder objective. For the decoder-objective ablation, we replace the ℓ_1 loss with $\ell_1 + \mathrm{LPIPS} + 0.5\,\mathcal{L}_{\mathrm{GAN}}$. LPIPS (Zhang et al., 2018) is applied to four randomly sampled frames per clip, and the adversarial term uses a three-layer PatchGAN discriminator (Isola et al., 2017) with hinge loss. The discriminator is enabled after 3,100 steps, with adaptive gradient-norm balancing following latent diffusion models (Rombach et al., 2022).

F DETAILS ON BASELINES

We compare JEPACast with representative deterministic, hybrid, and generative nowcasting approaches on SEVIR and MeteoNet. Below, we briefly summarize the modeling characteristics of the baselines reported in the main results tables.

Deterministic models.

- **ConvLSTM** (Shi et al., 2015) formulates precipitation nowcasting as spatiotemporal sequence forecasting and replaces the input-to-state and state-to-state transformations of an LSTM with convolutional operations.
- **ConvGRU** (Shi et al., 2017) is a convolutional recurrent baseline that uses GRU-style gating to model spatiotemporal evolution in radar sequences.
- **PredRNN** (Wang et al., 2017) introduces the Spatiotemporal LSTM (ST-LSTM), whose memory states propagate both through time and across stacked recurrent layers to jointly model spatial appearances and temporal variations.
- **PhyDNet** (Le Guen & Thome, 2020) uses a two-branch recurrent architecture that disentangles PDE-constrained physical dynamics from complementary unknown dynamics for future-frame prediction.
- **SimVP** (Gao et al., 2022a) is an end-to-end video prediction model built entirely from convolutional networks and trained with an MSE objective, without recurrent or attention-based prediction modules.
- **EarthFormer** (Gao et al., 2022b) is a space–time Transformer that models spatiotemporal dependencies using Cuboid Attention.
- **exPreCast** (Song et al., 2026) combines local spatiotemporal attention, a texture-preserving Cubic Dual Upsampling decoder, and a temporal extractor for precipitation nowcasting.

Hybrid models.

- **NowcastNet** (Zhang et al., 2023) combines a deterministic evolution network that models precipitation dynamics under advective conservation with a stochastic generative network conditioned on the predicted physical evolution.
- **CasCast** (Gong et al., 2024) cascades a deterministic pixel-space predictor for mesoscale precipitation distributions with a latent-space diffusion model for small-scale patterns.
- **DiffCast** (Yu et al., 2024) decomposes precipitation evolution into a deterministic trend and a stochastic residual modeled with diffusion. On MeteoNet, we report variants using ConvGRU, PhyDNet, and SimVP as the deterministic backbone.

Generative and probabilistic models.

- **LDM** (Rombach et al., 2022) performs diffusion in the compressed representation of a pretrained autoencoder, providing the generic latent-diffusion formulation used as a generative baseline.
- **STRPM** (Chang et al., 2022) is a high-resolution video prediction model that uses spatiotemporal residual prediction to preserve spatial detail and model temporal evolution.
- **MCVD** (Voleti et al., 2022) uses masked conditional video diffusion for probabilistic future-frame prediction by conditioning the denoising process on observed video frames.
- **PreDiff** (Gao et al., 2023) performs probabilistic spatiotemporal forecasting with conditional latent diffusion and introduces knowledge alignment using domain-specific constraints.
- **STLDM** (Foo et al., 2025) combines a deterministic forecasting stage from a conditioning network with a latent-diffusion enhancement stage, while learning its latent representation jointly with the VAE and conditioning network.
- **FlowCast** (Perrone Ribeiro & Faganeli Pucer, 2026) uses conditional flow matching to learn a direct noise-to-data generative process in a compressed latent space for probabilistic precipitation forecasting.
- **FREUD** (Schusterbauer et al., 2026) uses a frame-wise encoder and a generative rectified-flow decoder for uncertainty-preserving compression, together with a probabilistic forecasting model operating in the resulting latent space.

Evaluation resolution. For SEVIR, all models are evaluated at the native 384×384 resolution. LDM, PreDiff, and STLDM, marked by † in Table 2, are trained on 128×128 downsampled data and evaluated after upsampling their predictions to 384×384. For MeteoNet, we evaluate all methods at 128×128 resolution.

G Evaluation Metrics

We evaluate forecasts using threshold-based meteorological skill scores together with continuous image- and distribution-level metrics. For a precipitation threshold τ, let $\widehat{y}_i^\tau = \mathbb{1}[\widehat{X}_i \geq \tau]$ and $y_i^\tau = \mathbb{1}[X_i \geq \tau]$. We define the contingency counts as

$$\begin{aligned} H_\tau &= \sum_i \widehat{y}_i^\tau y_i^\tau, & M_\tau &= \sum_i (1-\widehat{y}_i^\tau) y_i^\tau, \\ F_\tau &= \sum_i \widehat{y}_i^\tau (1-y_i^\tau), & C_\tau &= \sum_i (1-\widehat{y}_i^\tau)(1-y_i^\tau), \end{aligned} \tag{10}$$

where H, M, F, and C denote hits, misses, false alarms, and correct rejections, respectively.

Critical Success Index (CSI). CSI measures the overlap between predicted and observed precipitation events while ignoring correct rejections:

$$\text{CSI-}\tau = \frac{H_\tau}{H_\tau + M_\tau + F_\tau}. \tag{11}$$

Higher values indicate better forecast skill. For the dataset-specific threshold set $\mathcal{T}$, we report the mean

$$\text{CSI-M} = \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} \text{CSI-}\tau, \tag{12}$$

using the six SEVIR thresholds and four MeteoNet thresholds defined in Appendix D. For SEVIR, we additionally report CSI-219, corresponding to the highest threshold, to assess forecast skill for high-intensity precipitation.

Spatially pooled CSI. Pixel-wise CSI strongly penalizes small spatial displacement errors. To additionally assess neighborhood-scale forecast skill on SEVIR, we apply spatial max pooling before computing CSI. For pooling scale $s \in \{4, 16\}$,

$$\mathcal{P}_s(\mathbf{X})_{i,j} = \max_{(u,v) \in \mathcal{N}_s(i,j)} X_{u,v}, \tag{13}$$

where $\mathcal{N}_s(i,j)$ denotes an $s \times s$ spatial neighborhood. We report both CSI-M and CSI-219 after 4×4 (P4) and 16×16 (P16) pooling.

Heidke Skill Score (HSS). HSS evaluates categorical forecast skill relative to that expected from random forecasts with the same marginal frequencies:

$$\text{HSS-}\tau = \frac{2(H_\tau C_\tau - M_\tau F_\tau)}{(H_\tau + M_\tau)(M_\tau + C_\tau) + (H_\tau + F_\tau)(F_\tau + C_\tau)}. \tag{14}$$

A score of 1 indicates perfect prediction, while 0 corresponds to no improvement over chance under the contingency-table formulation. As with CSI, we report the mean over the dataset-specific threshold set,

$$\text{HSS-M} = \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} \text{HSS-}\tau. \tag{15}$$

Structural Similarity Index (SSIM). SSIM evaluates structural similarity between the predicted and observed radar fields:

$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{16}$$

where μ, σ^2, and σ_{xy} denote local means, variances, and covariance. Higher SSIM indicates greater structural similarity.

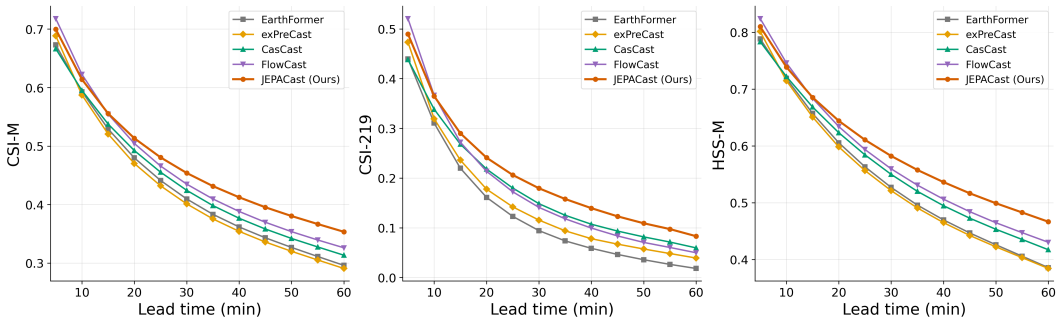

Figure 9: **Forecast skill across lead times on SEVIR.** CSI-M, CSI-219, and HSS-M are reported at each forecast lead time for representative deterministic, hybrid, and generative baselines. Performance is comparable among the strongest methods at short lead times, while JEPACast shows clearer separation toward longer horizons.

Continuous Ranked Probability Score (CRPS). For SEVIR, we follow the Gaussian CRPS evaluation used by prior work (Schusterbauer et al., 2026). At each pixel and lead time, let μ and σ denote the mean and standard deviation of the forecast ensemble, and let $z = (X - \mu)/\sigma$. The CRPS of the corresponding Gaussian predictive distribution is

$$\mathrm{CRPS} = \sigma \left[z\big(2\Phi(z) - 1\big) + 2\phi(z) - \frac{1}{\sqrt{\pi}} \right], \tag{17}$$

where ϕ and Φ are the standard normal density and cumulative distribution functions, respectively; for $\sigma = 0$, the expression is understood in its limiting form $|\mu - X|$. Lower values indicate better probabilistic forecasts. We average CRPS over spatial locations, forecast lead times, and test samples.

Final-lead CSI. To explicitly measure degradation at long lead times, we additionally report CSI-M at the final predicted frame, corresponding to a lead time of 60 minutes on SEVIR in Figure 1.

H Additional Analysis

Performance across lead times. Figure 9 compares forecast skill across the 60-minute prediction horizon on SEVIR. At short lead times, the strongest methods achieve comparable performance, while JEPACast shows increasingly clear separation toward intermediate and longer forecast horizons. The trend is particularly visible for CSI-219 and HSS-M, consistent with the stronger longer-horizon performance observed in the aggregate results.

CFG weight and number of sampling steps. Figure 10 sweeps the two sampler choices of the SEVIR model: $\{1, 2, 3, 4, 6, 10\}$ Euler steps with no guidance, constant guidance weights $w \in \{1.5, 2, 2.5, 3\}$, and a lead-time ramp from 1.25 to 2.0. For this analysis, we evaluate the first 2,048 test windows using 10 ensemble members, with noise draws shared across configurations such that member k always starts from the same seed.

Across guided configurations, both CSI-M and CSI-219 peak at two Euler steps and decrease with further sampling, even though additional steps consistently reduce ensemble CRPS and produce sharper individual members. This indicates that improved sample quality does not necessarily translate into better threshold-based skill of the ensemble mean. The effect is particularly pronounced with strong guidance at a single Euler step, where increasing w sharply degrades CSI. The two-step sampler with ramped guidance achieves the strongest CSI performance, whereas increasing to ten steps incurs roughly $5\times$ the sampling cost while yielding lower CSI at both moderate and high precipitation thresholds.

Figure 11 further examines the effect of guidance across forecast lead times with two-step sampling fixed. Without guidance, the ensemble mean is approximately calibrated for heavy precipitation at short lead times but increasingly underpredicts VIL$\geq$219 as the forecast horizon grows. A strong

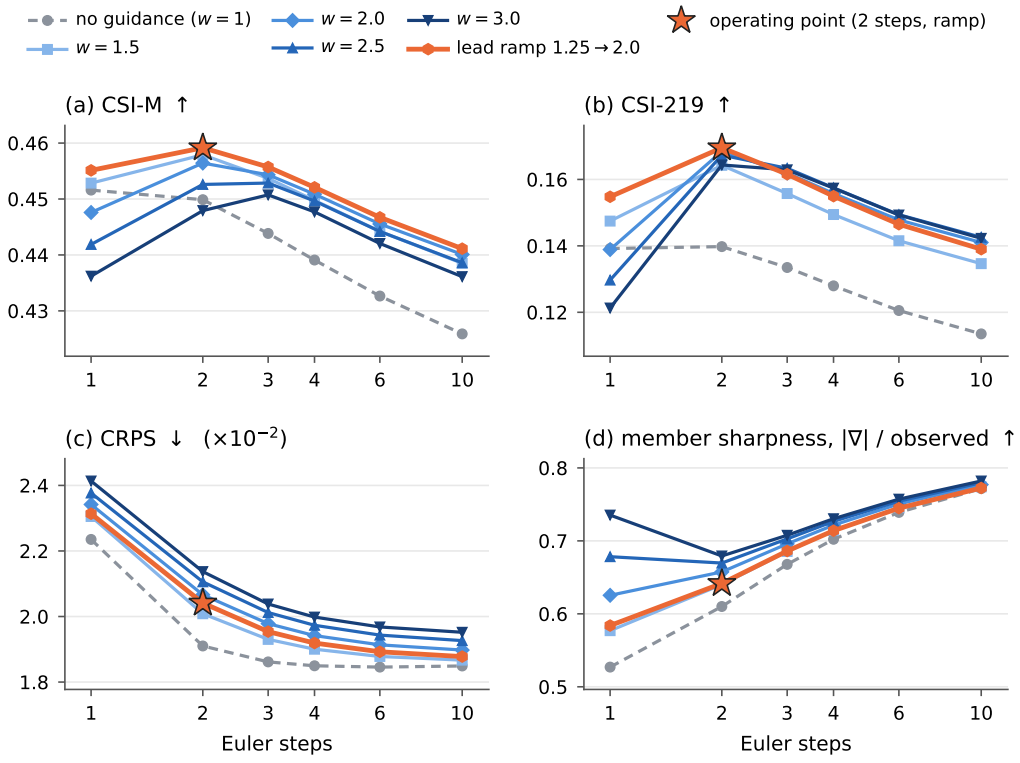

Figure 10: **Euler steps × guidance on the SEVIR model**. Each graph denotes (a) CSI-M, (b) CSI-219, (c) CRPS of the ensemble, and (d) member sharpness, mean $|\nabla|$ relative to the observation.

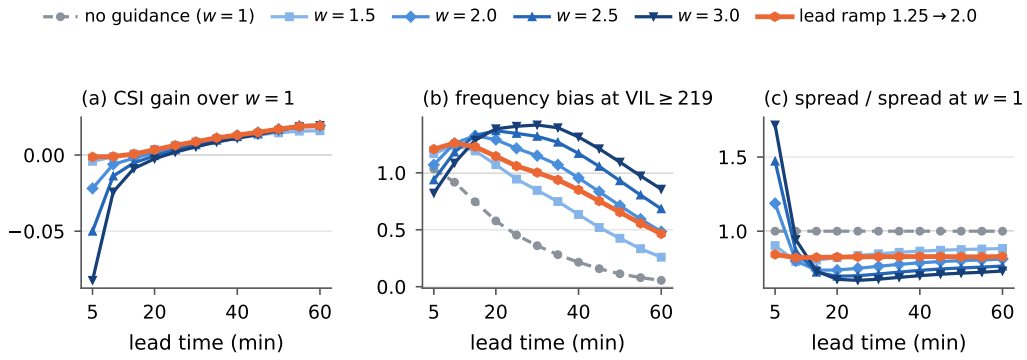

Figure 11: **Changing guidance weights with fixed two-step sampling.** (a) CSI gain over no guidance, (b) frequency bias of the ensemble mean at VIL≥219 (1 = the forecast covers the observed area), (c) member spread relative to no guidance.

constant guidance weight compensates for this long-range bias, but overcorrects the early forecasts and reduces CSI at short lead times. Ramping the guidance weight from 1.25 to 2.0 instead follows the increasing correction required over the forecast horizon, keeping the frequency bias closer to one while avoiding the early-lead degradation of stronger constant guidance. The ramp also retains more member spread than the larger fixed weights. Across the evaluated configurations, it achieves the best overall CSI and CSI-219, with a paired improvement of 0.001–0.0015 over the strongest fixed-weight setting ($w = 1.5$), whose bootstrap confidence interval excludes zero. Together, these results characterize why our default operating point uses two Euler steps with lead-time-ramped guidance.

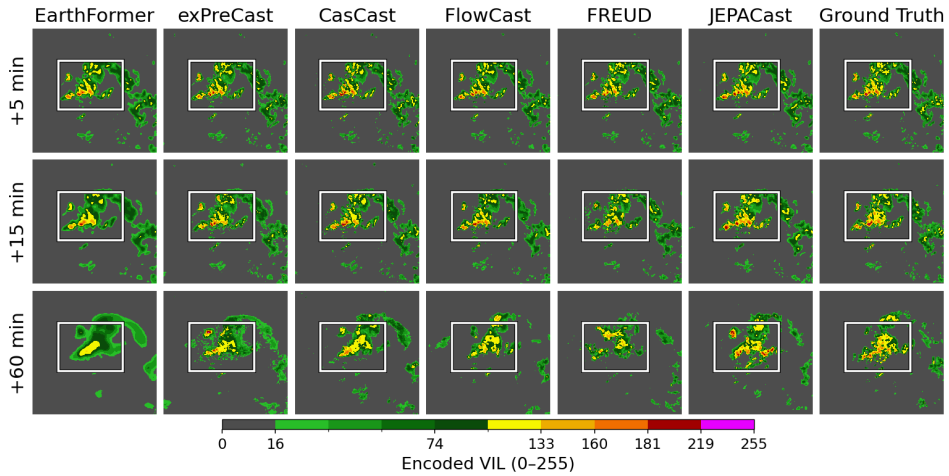

Figure 12: **Additional qualitative results on SEVIR.** Forecasts from EarthFormer, exPreCast, CasCast, FlowCast, FREUD, and JEPACast are compared with the ground truth at lead times of $+5$, $+15$, and $+60$ minutes. Boxed regions highlight representative local structures for closer comparison. Colors indicate encoded VIL intensity.

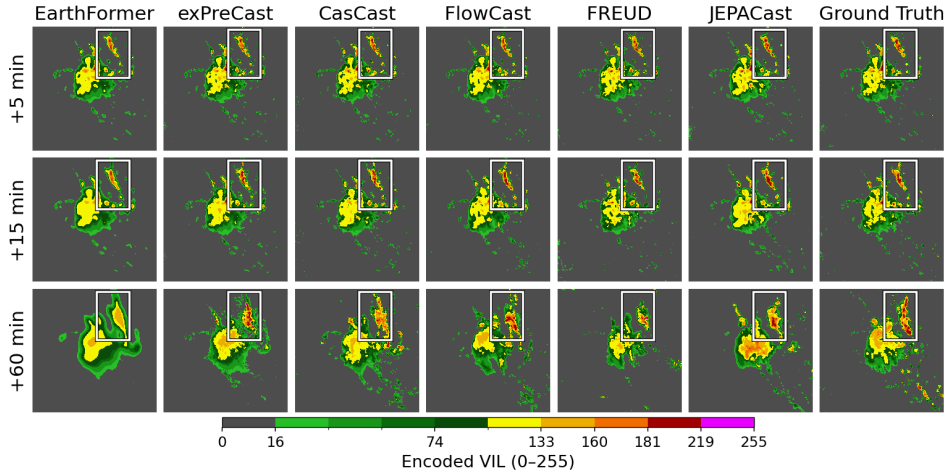

Figure 13: **Additional qualitative results on SEVIR.** Forecasts from EarthFormer, exPreCast, CasCast, FlowCast, FREUD, and JEPACast are compared with the ground truth at lead times of $+5$, $+15$, and $+60$ minutes. Boxed regions highlight representative local structures for closer comparison. Colors indicate encoded VIL intensity.

I Additional Qualitative Results

Additional forecast examples. Figures 12 and 13 provide additional qualitative comparisons on SEVIR, extending the examples shown in Figure 4. We compare the same deterministic, hybrid, and generative baselines at short, intermediate, and long lead times. The examples illustrate differences in localization, spatial structure, and precipitation intensity as the forecast horizon increases.

Extended longer lead-time rollout. Figure 14 extends the longer-horizon comparison in Figure 5 to $+240$ minutes by recursively generating 60-minute forecast blocks without additional ground-truth input. In the illustrated sequence, JEPACast retains more localized precipitation structures while FlowCast produces smoother forecasts as the rollout horizon increases. This comparison illustrates the behavior of the two latent forecasting spaces beyond the 60-minute training horizon.

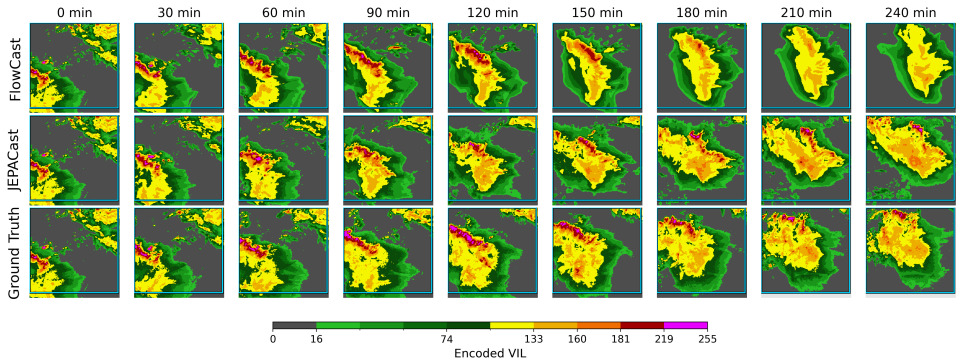

Figure 14: **Extended longer lead-time rollout on SEVIR.** Conditioned on 13 observed frames, FlowCast and JEPACast recursively forecast up to $+240$ minutes without further ground-truth input. Ground-truth coverage is extended by aligning two overlapping SEVIR events in space and time. Cyan boxes mark their shared region shown for comparison; light-gray areas beyond $+180$ minutes indicate unavailable ground truth. Colors represent encoded VIL intensity.

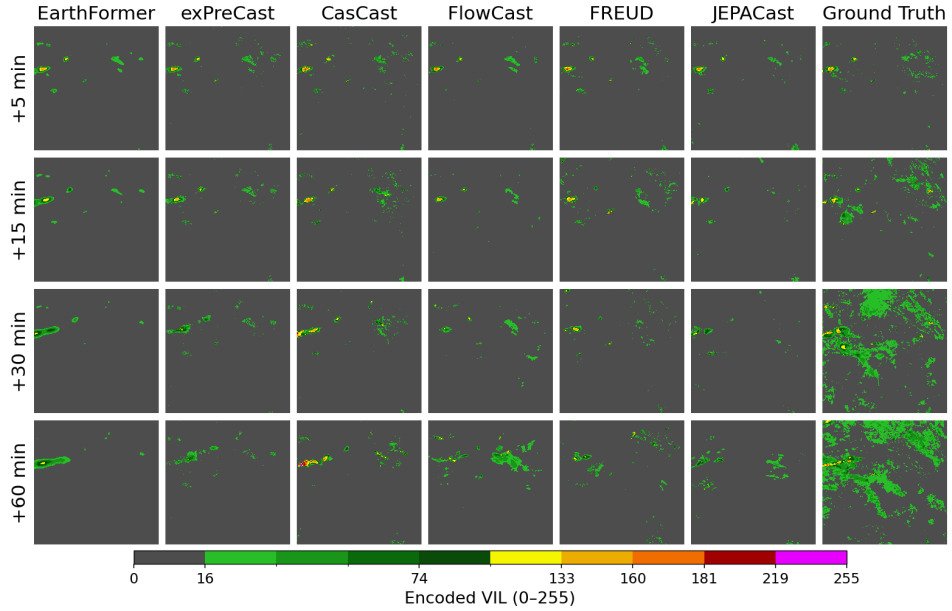

Figure 15: **Failure case on SEVIR.** The example highlights a challenging case in which all evaluated methods underpredict precipitation initiation and subsequent areal growth. Colors indicate encoded VIL intensity.

Failure case analysis. Figure 15 shows a shared failure mode under rapidly developing precipitation. While all methods reasonably extrapolate the sparse echoes observed at short lead times, they substantially underpredict the broad initiation and areal growth that emerges in the ground truth after $+15$ minutes. The discrepancy becomes pronounced by $+30$ and $+60$ minutes, where precipitation growth dominates the observed evolution. This illustrates a limitation of radar-only learned nowcasting: precipitation development that is only weakly expressed in the observed radar history remains difficult to anticipate.

J Broader Impacts

Reliable short-range precipitation forecasts can support weather-sensitive decision-making, including flood response, transportation management, and severe-weather monitoring. Because localized

heavy rainfall can evolve rapidly, improvements in spatially resolved nowcasting may provide useful information for anticipating hazardous conditions and supporting timely responses. The approach studied in this work contributes to this broader objective by improving the modeling of near-term precipitation evolution, particularly for intense rainfall and longer lead times.

Nevertheless, JEPACast is a research model and should not be interpreted as a replacement for operational forecasting systems or human expertise. In practical settings, model predictions should be considered together with numerical weather prediction, observational data, and the judgment of trained meteorologists and other domain experts. Further validation across geographic regions, sensing systems, and rare extreme events, together with appropriate uncertainty communication and expert oversight, would be necessary before deployment in safety-critical applications.